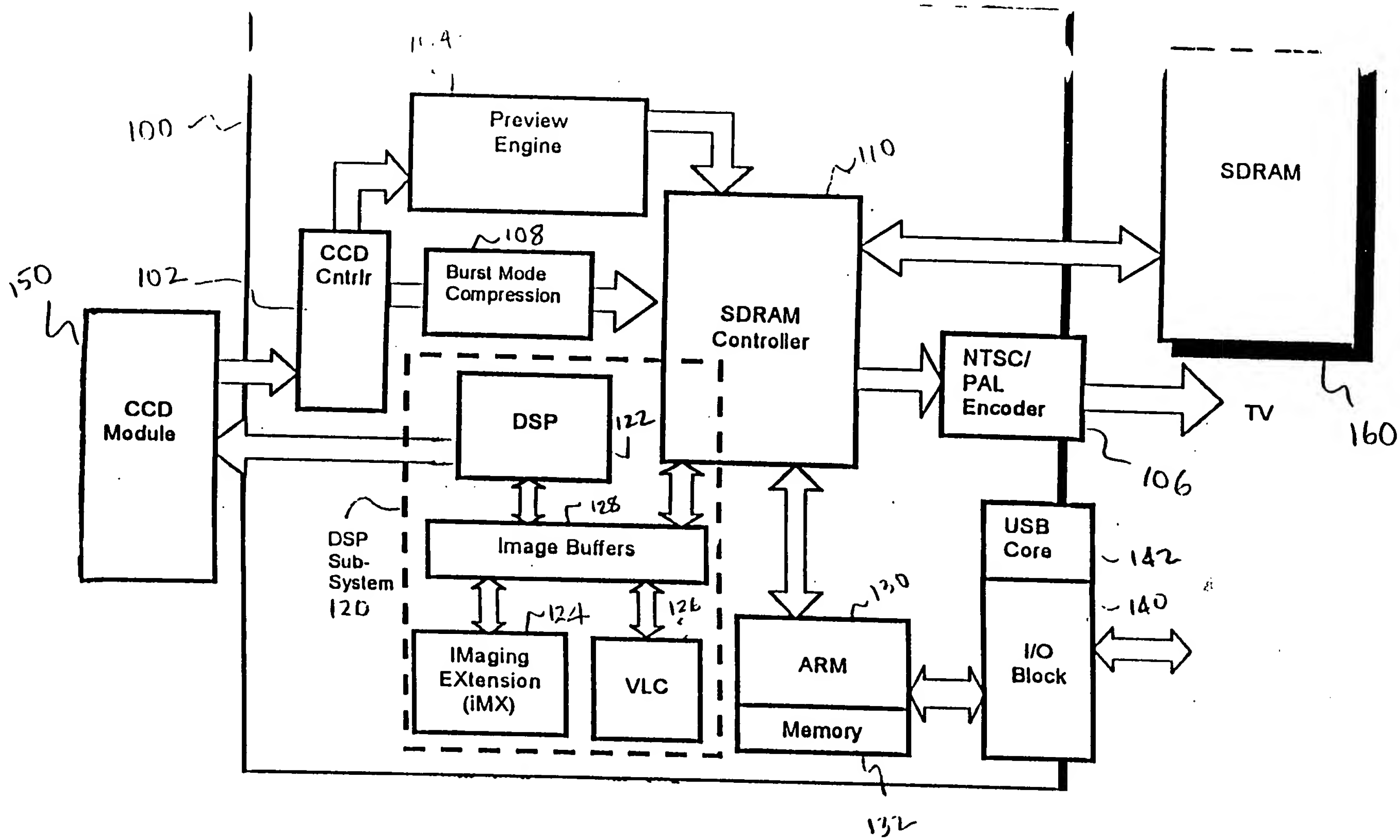


100
150
102
114
CCD Module
CCD Cntrlr
Preview Engine
108
Burst Mode Compression
110
SDRAM Controller
SDRAM
160
NTSC/ PAL Encoder
TV
106
DSP
122
128
Image Buffers
DSP Sub-System 120
124
126
IMaging EXtension (iMX)
VLC
130
ARM
Memory
132
USB Core
142
140
I/O Block

Fig. 1a

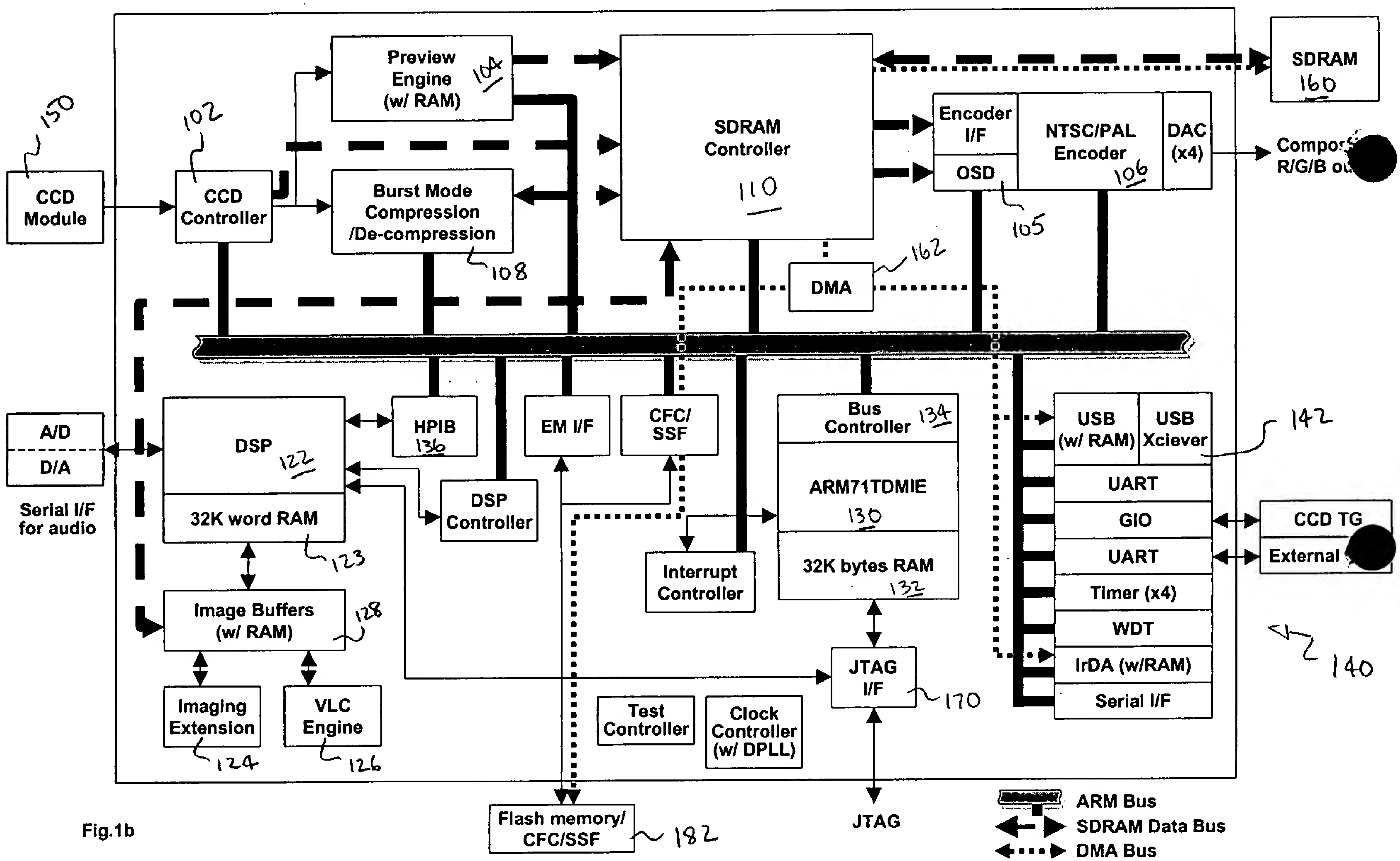

CCD Module
150
CCD Controller
102
Preview Engine (w/ RAM)
104
Burst Mode Compression /De-compression
108
SDRAM Controller
110
DMA
162
Encoder I/F
OSD
105
NTSC/PAL Encoder
106
DAC (x4)
Compos R/G/B ou
SDRAM
160
A/D
D/A
Serial I/F for audio
DSP
122
32K word RAM
123
HPIB
136
DSP Controller
EM I/F
CFC/ SSF
Image Buffers (w/ RAM)
128
Imaging Extension
124
VLC Engine
126
Interrupt Controller
Bus Controller
134
ARM71TDMIE
130
32K bytes RAM
132
JTAG I/F
170
JTAG
Test Controller
Clock Controller (w/ DPLL)
Flash memory/ CFC/SSF
182
USB (w/ RAM)
USB Xciever
142
UART
GIO
UART
Timer (x4)
WDT
IrDA (w/RAM)
Serial I/F
CCD TG
External
140
ARM Bus
SDRAM Data Bus
DMA Bus

Fig.1b

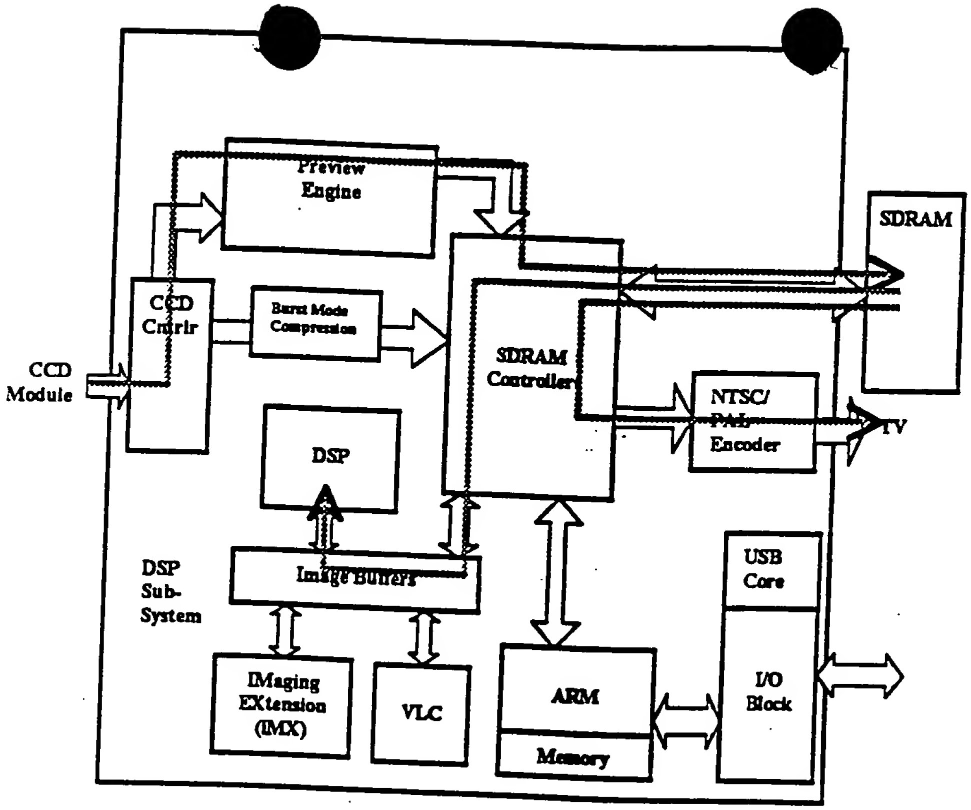


Figure 2 Data flow in the preview mode

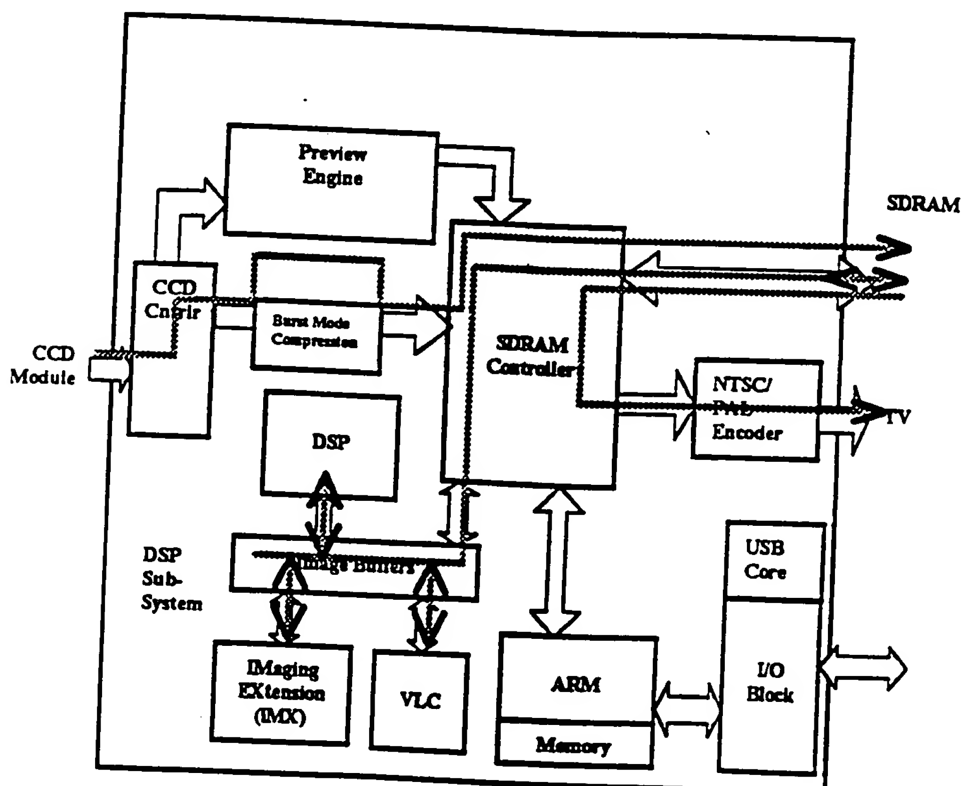


Figure 3 Data flow in the Capture Mode

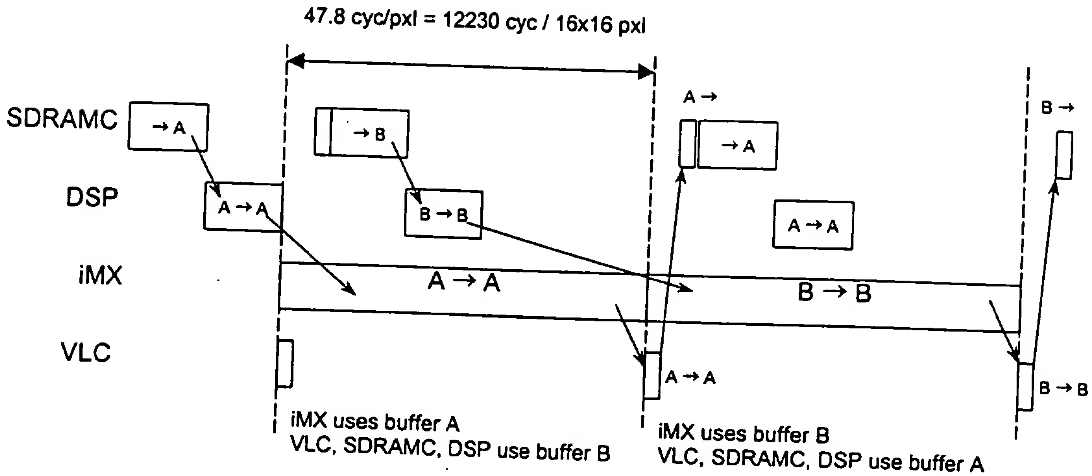
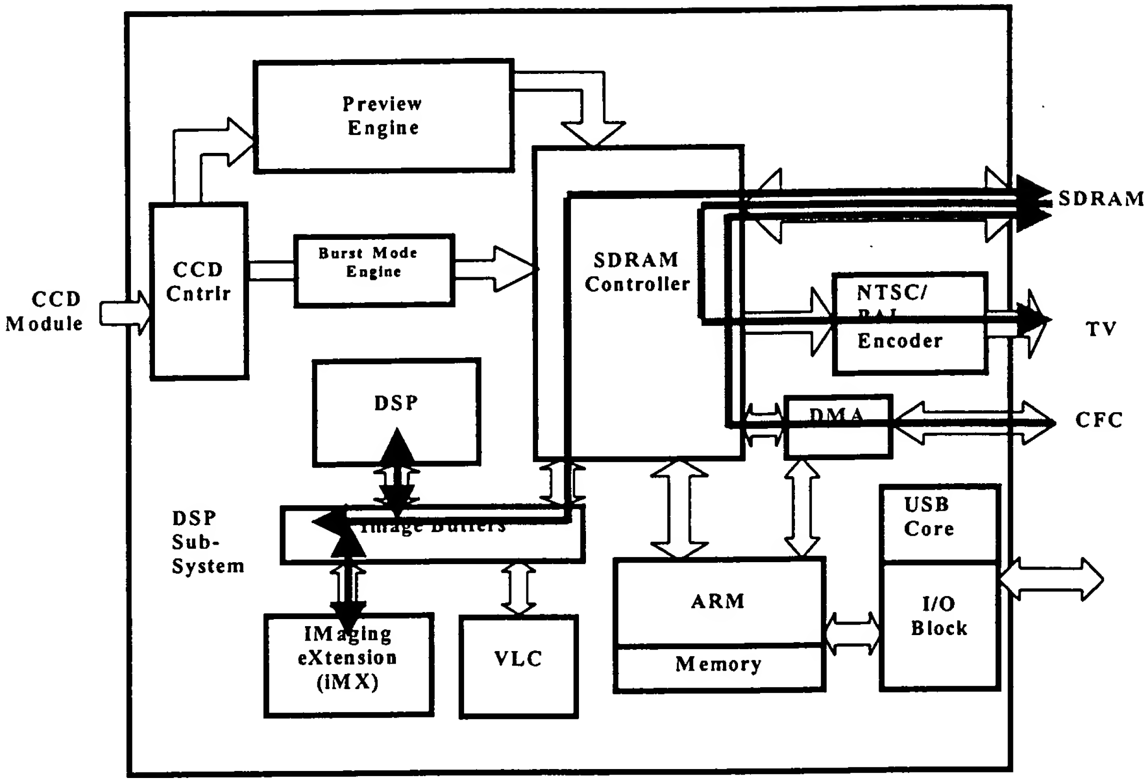


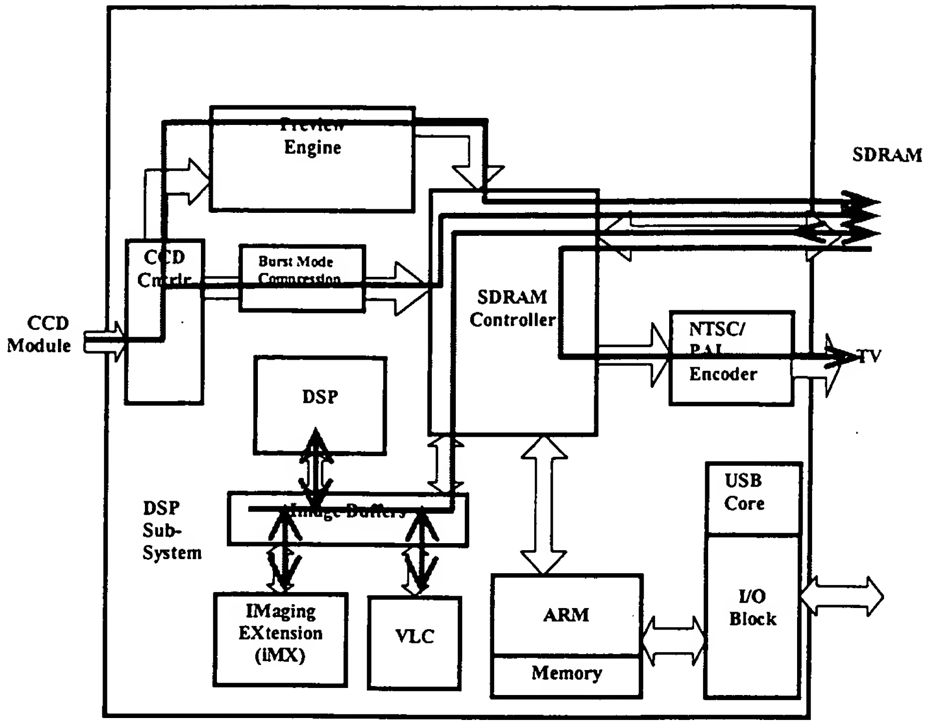
Figure 4 Capture mode schedule, with DSP, iMX, VLC, and SDRAMC
3b

Playback Mode

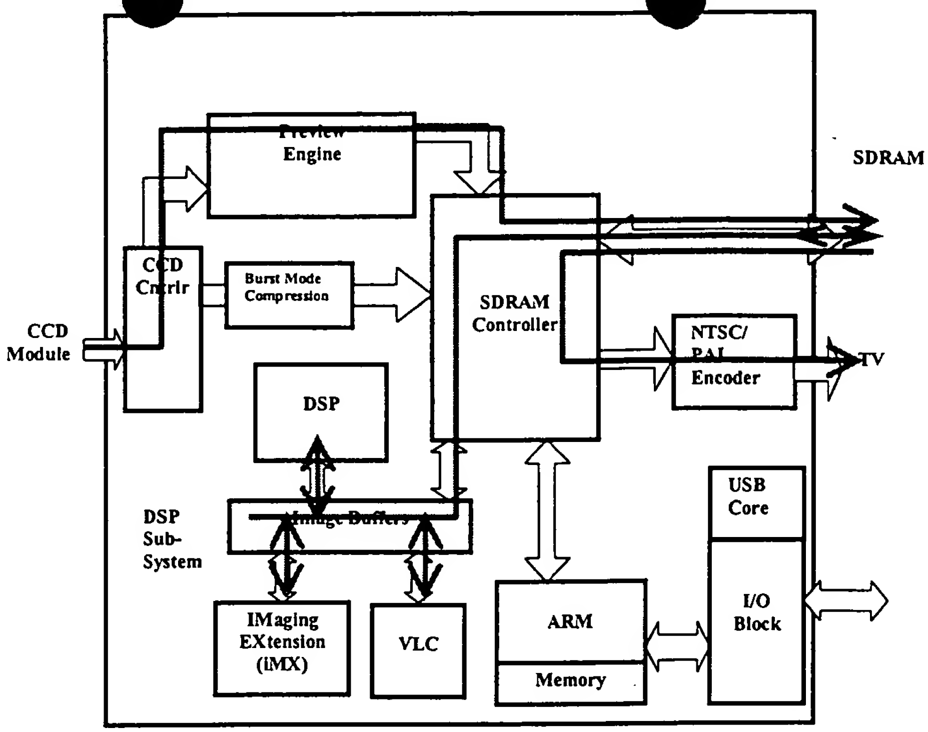


• Figure 4 Dataflow in the Playback mode

Burst Capture Mode



• Figure 5 Dataflow in the Burst Mode Capture



• Figure 6 Dataflow in the Burst Mode data processing offline

CFA Arrangements

R	G	R	G
G	B	G	B
R	G	R	G
G	B	G	B

Ye	Cy	Ye	Cy
G	Mg	G	Mg
Ye	Cy	Ye	Cy
G	Mg	G	Mg

7 (a) RGB bayer   7 (b) Complementary

Figure ~~4~~ 7 CFA arrangements supported by the Preview Engine

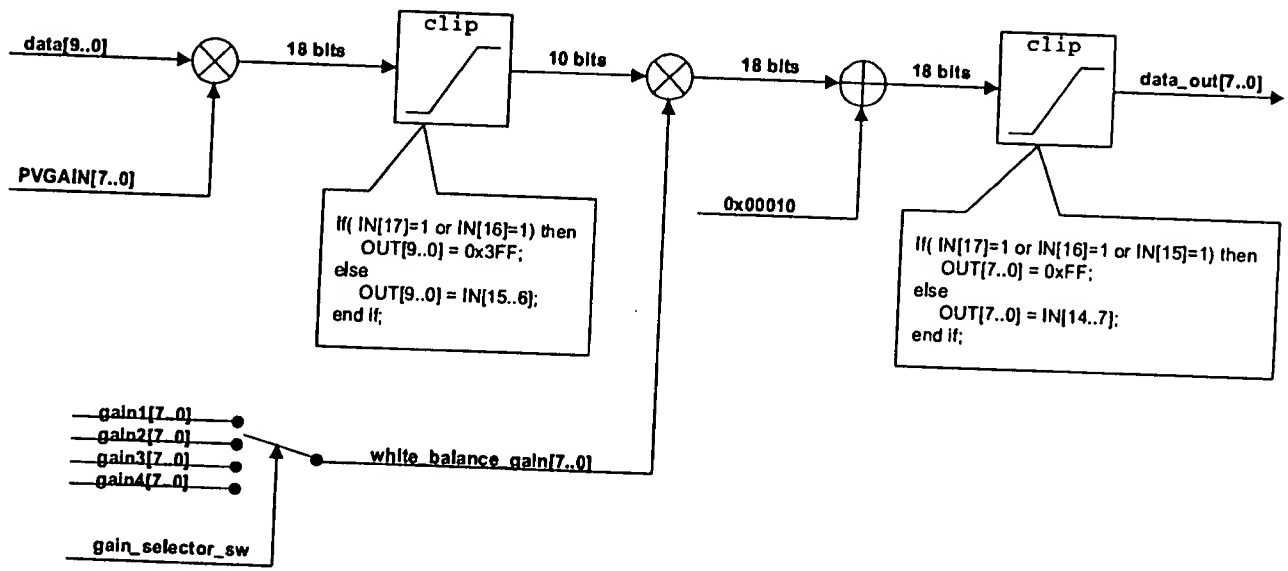


Fig ~~21~~ 8 Block diagram of white balance module

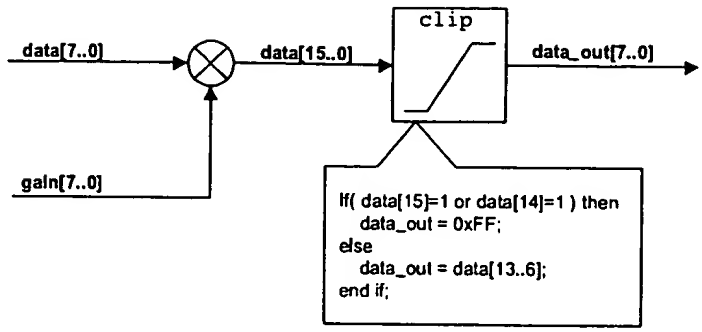


Fig 9a Block diagram of RGB gain for comp-CCD module

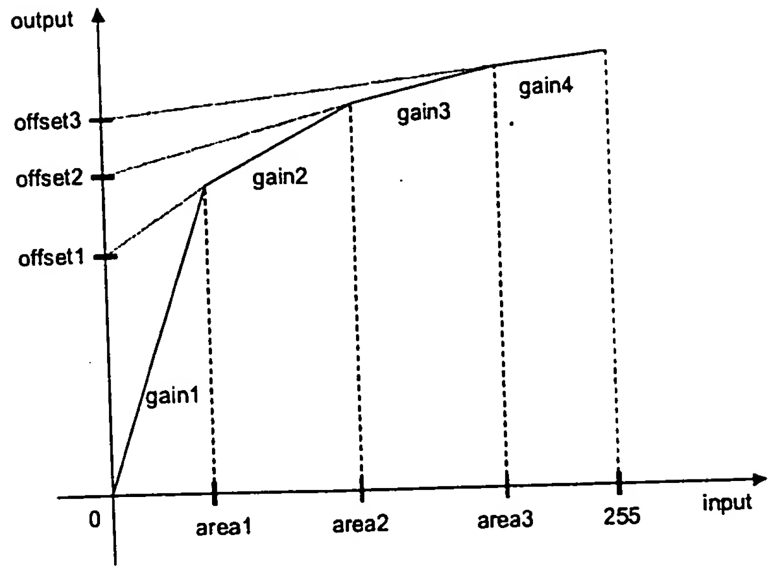


Fig. ~~2-4(a)~~ 9b Input data for gamma correction

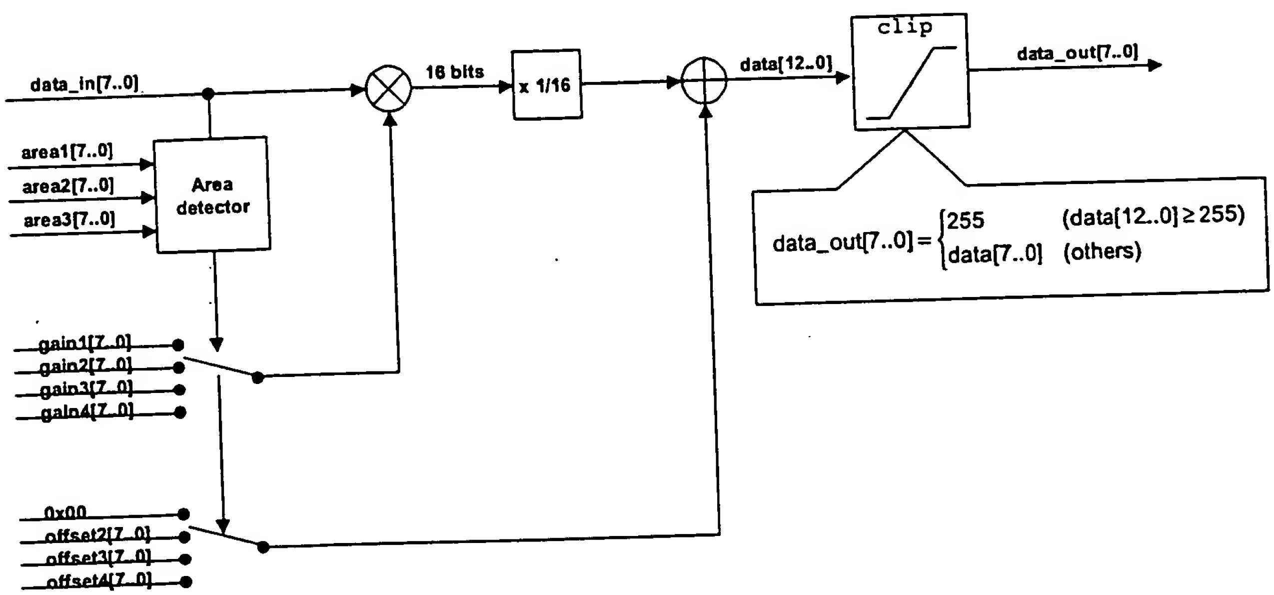


Fig ~~2-4(b)~~ 9c Block diagram of gamma correction module

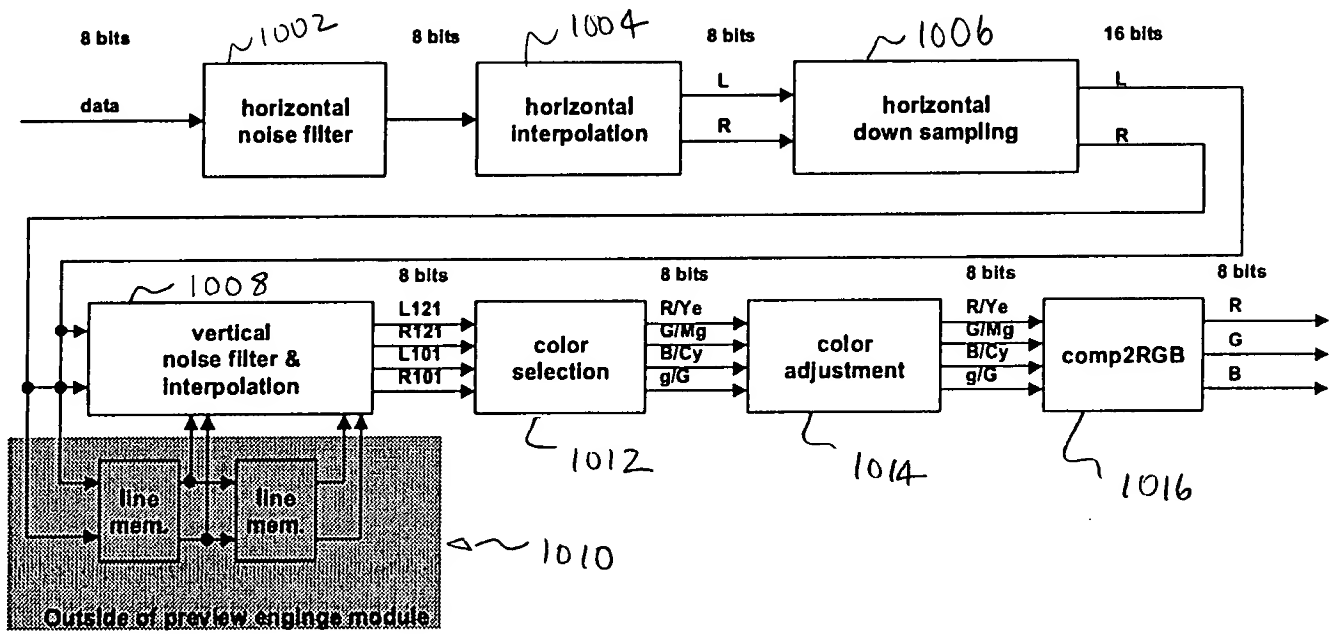


Fig 10a Block diagram of CFA interpolation module

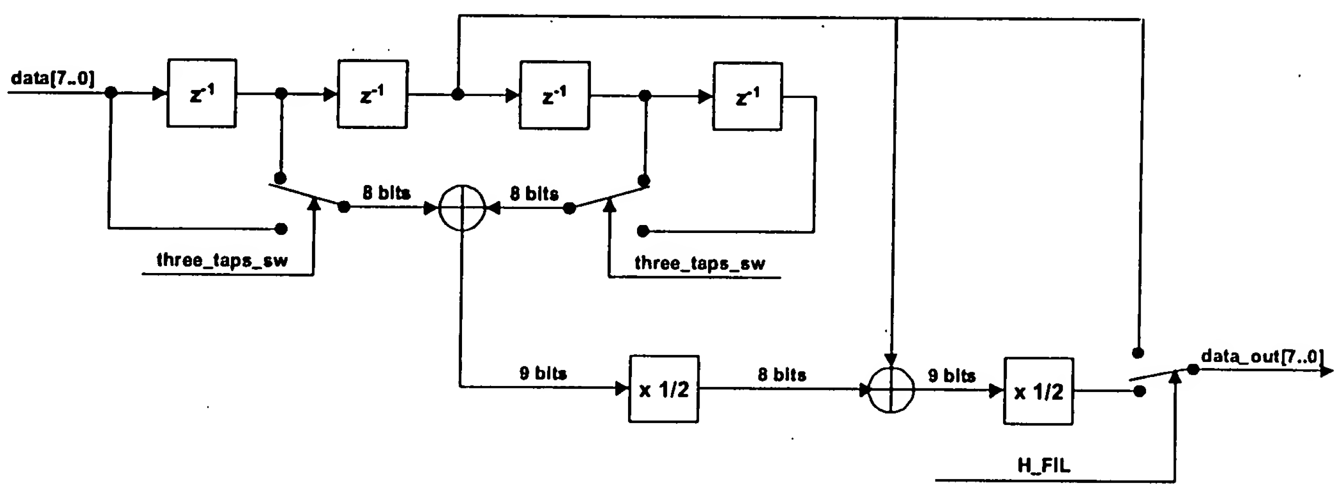


Fig 10b Block diagram of horizontal noise filter sub-module

Bayer CCD mode. In this figure, interpolated

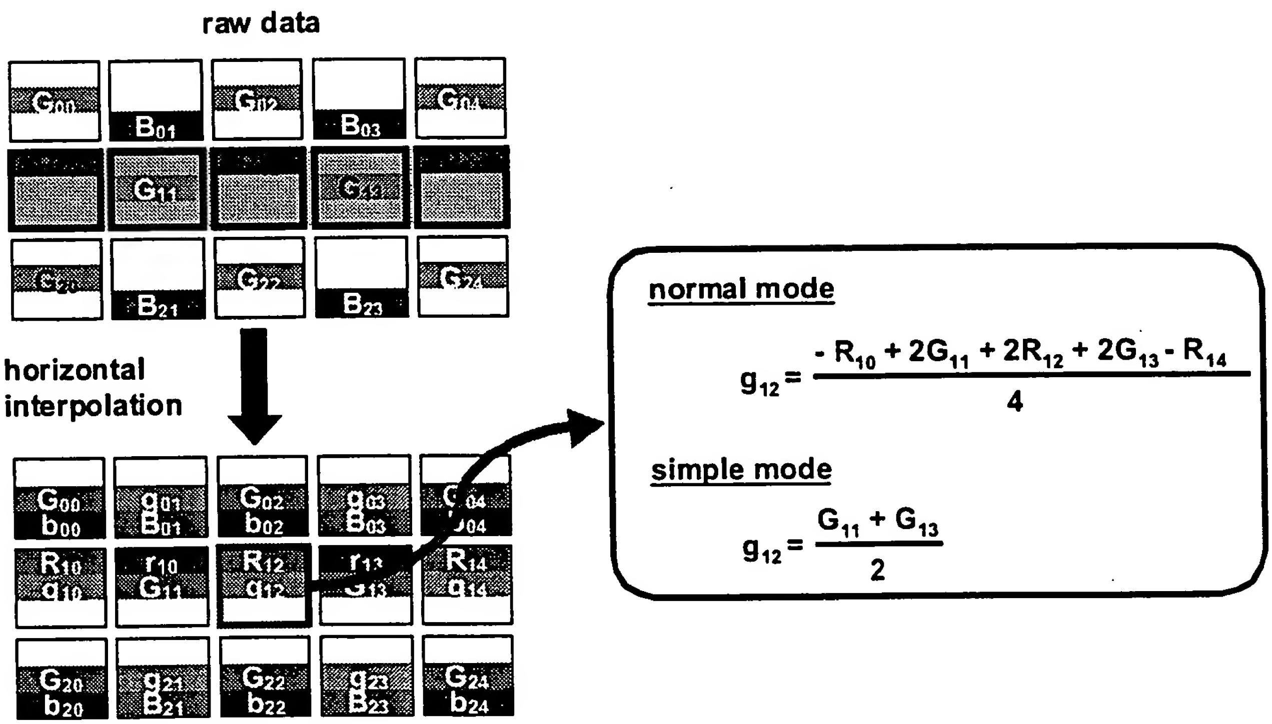


Fig ~~22(b)~~ 10c Example of horizontal interpolation processing (RGB Bayer)

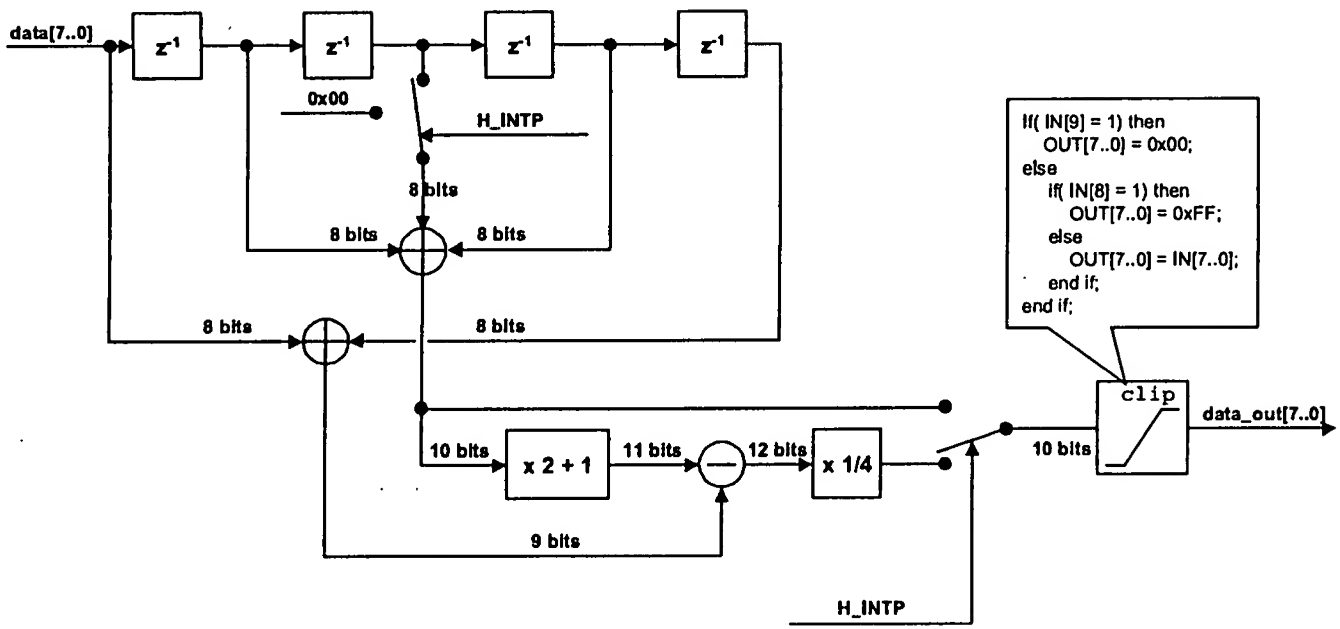


Fig 10d Block diagram of horizontal interpolation sub-module

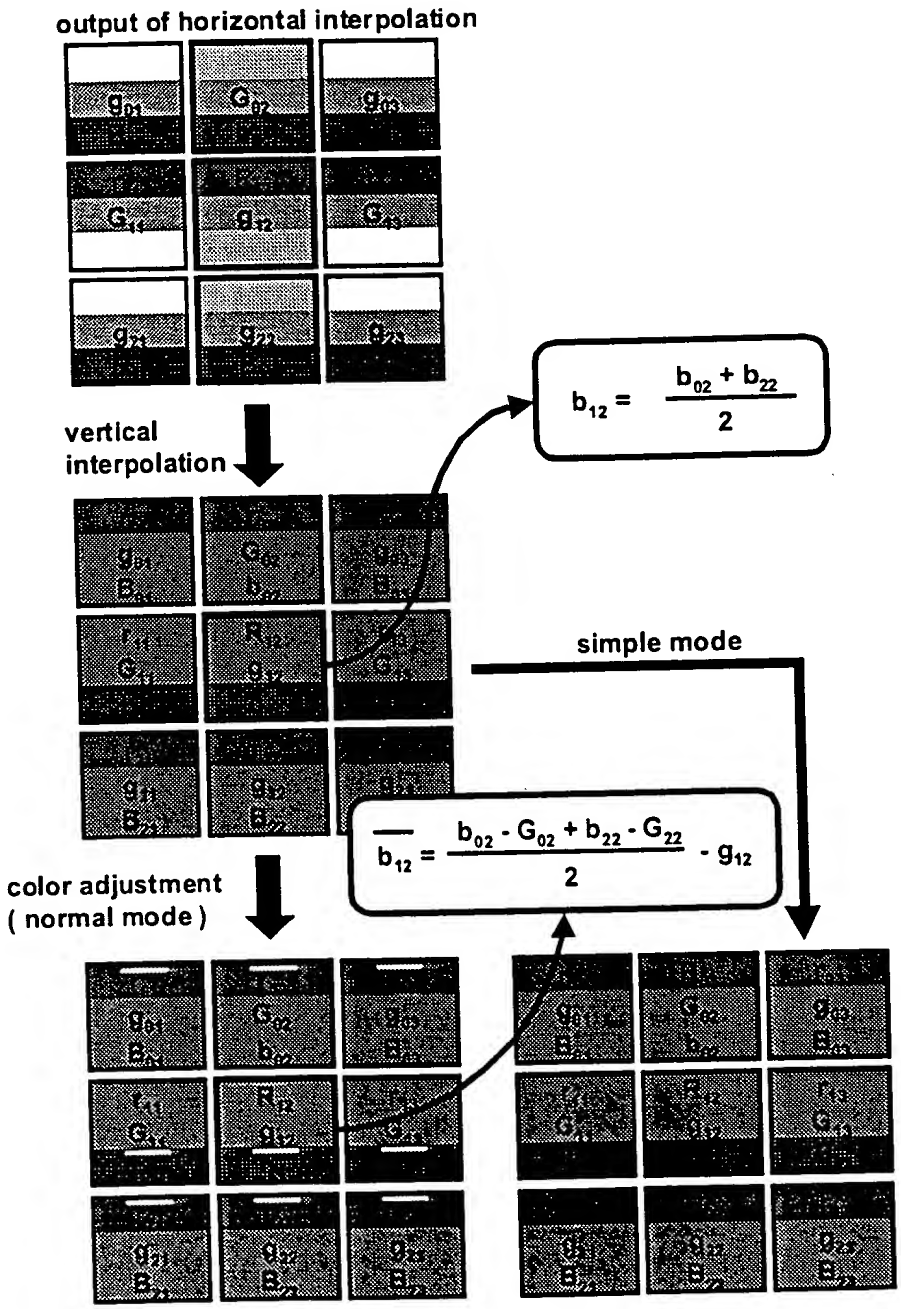


Fig 10e Example of vertical interpolation sequence(RGB Bayer)

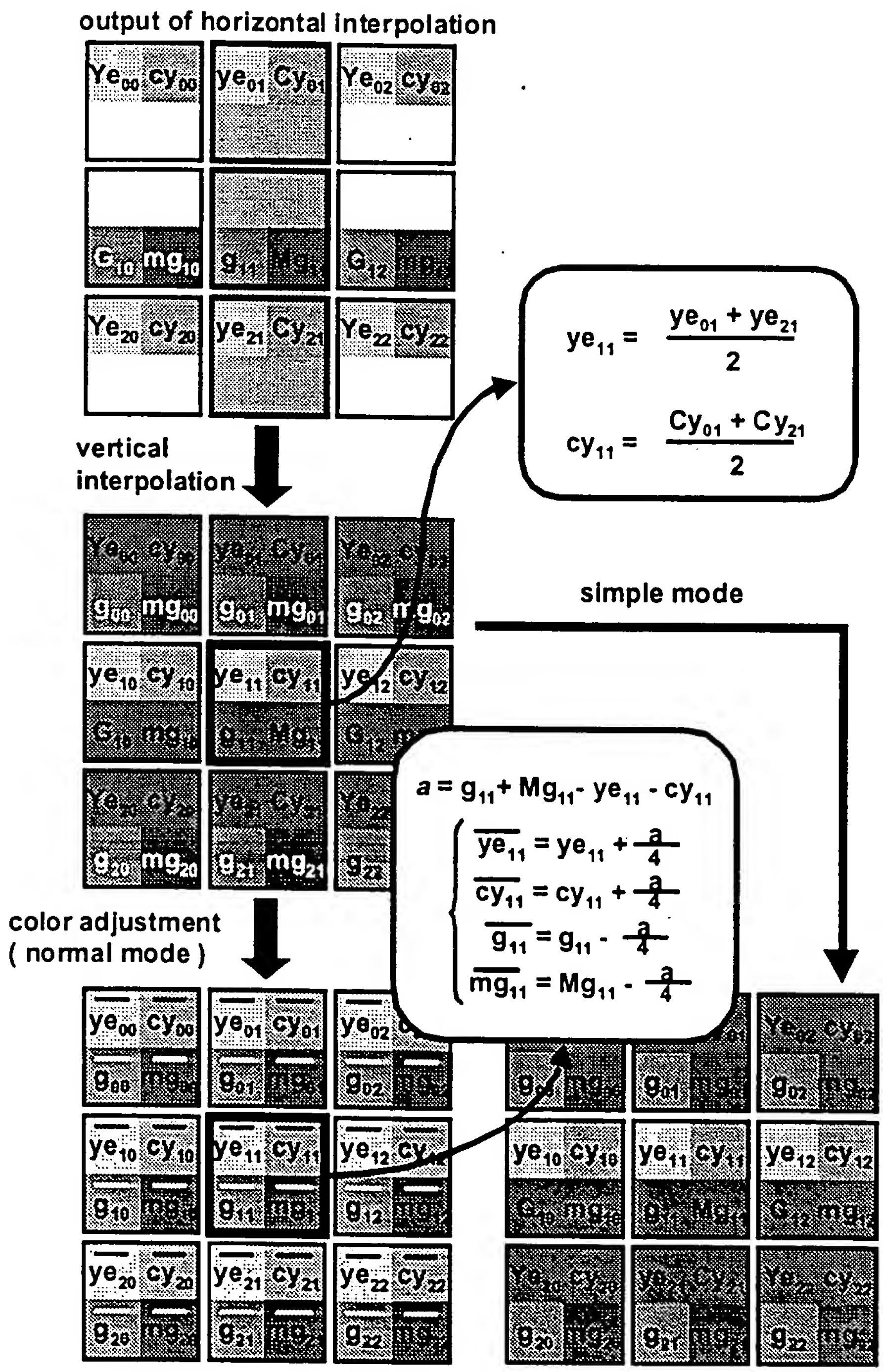
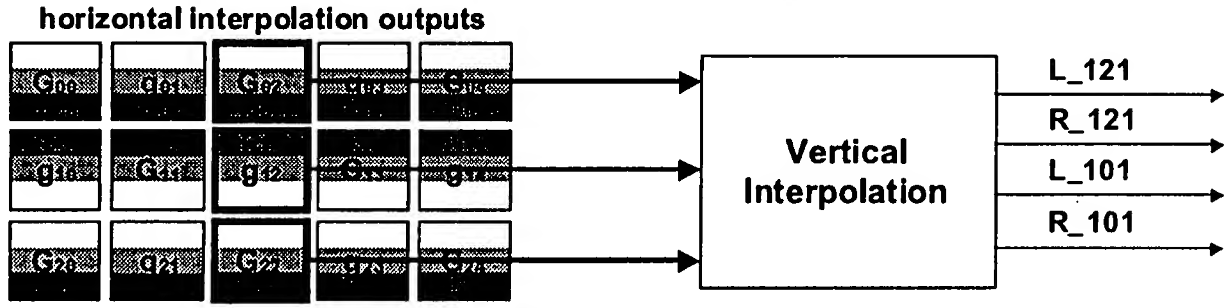


Fig 10f Example of vertical interpolation sequence (complementary CCD)



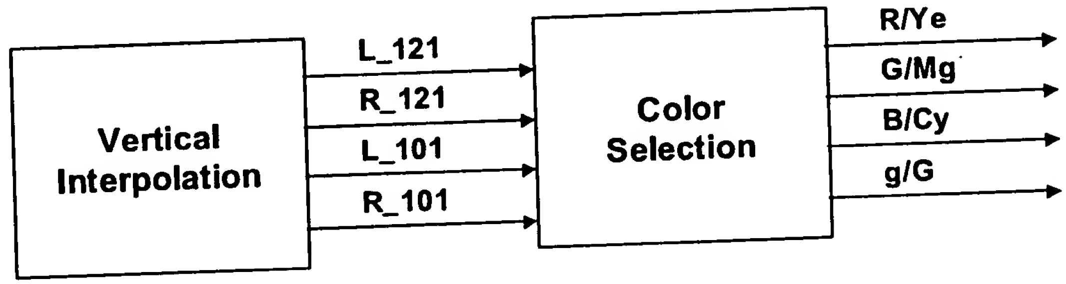
<u>noise filter = off</u>

$$\begin{cases} L\_121 = R_{12} \\ R\_121 = g_{12} \\ L\_101 = \dfrac{G_{02} + G_{22}}{2} \\ R\_101 = \dfrac{b_{02} + b_{22}}{2} \end{cases}$$

<u>noise filter = on</u>

$$\begin{cases} L\_121 = R_{12} - g_{12} + \dfrac{G_{02} + 2g_{12} + G_{22}}{4} \\ R\_121 = \dfrac{G_{02} + 2g_{12} + G_{22}}{4} \\ L\_101 = \dfrac{G_{02} + G_{22}}{2} \\ R\_101 = \dfrac{b_{02} + b_{22}}{2} \end{cases}$$

Fig 10g Example of outputs of vertical interpolation sub-module(RGB Bayer)



noise filter = off

$$\begin{cases} R/Ye = R_{12} \\ G/Mg = g_{12} \\ g/Cy = \dfrac{G_{02} + G_{22}}{2} \\ B/Cy = \dfrac{b_{02} + b_{22}}{2} \end{cases}$$

noise filter = on

$$\begin{cases} R/Ye = R_{12} - g_{12} + \dfrac{G_{02} + 2g_{12} + G_{22}}{4} \\ G/Mg = \dfrac{G_{02} + 2g_{12} + G_{22}}{4} \\ g/Cy = \dfrac{G_{02} + G_{22}}{2} \\ B/Cy = \dfrac{b_{02} + b_{22}}{2} \end{cases}$$

Fig ~~22(a)~~ 10i Example of color selection processing (RGB Bayer)

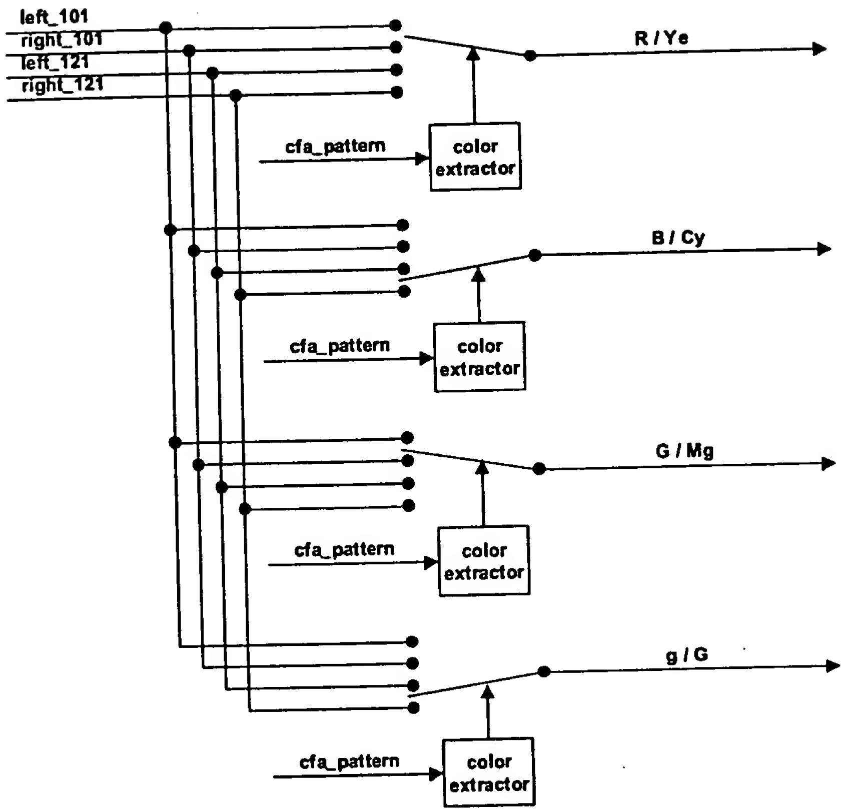
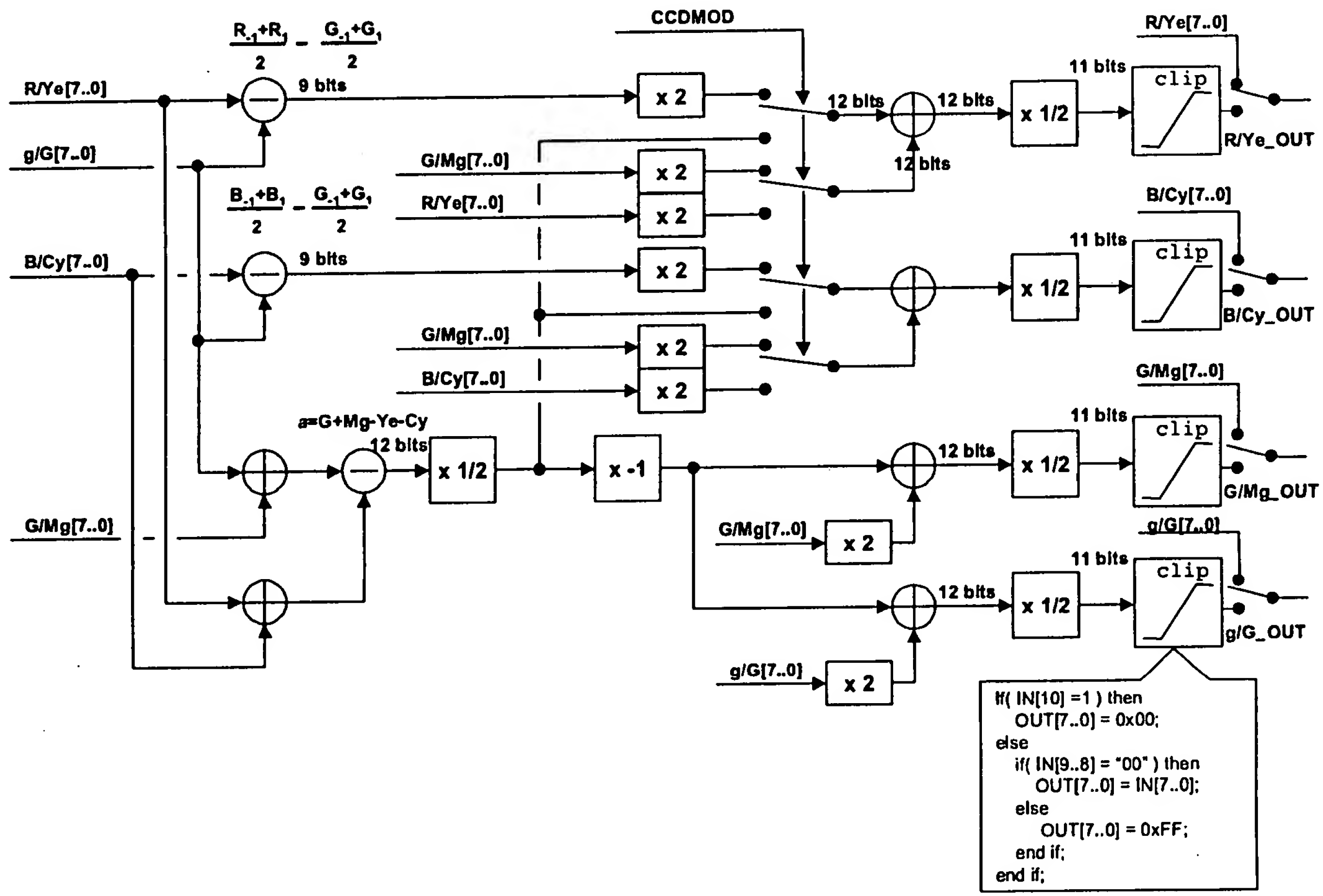


Fig 10j) Block diagram of color selection module



$\frac{R_{-1}+R_1}{2} - \frac{G_{-1}+G_1}{2}$

$\frac{B_{-1}+B_1}{2} - \frac{G_{-1}+G_1}{2}$

10k

Fig ~~22/28~~ **Block diagram of color adjustment module**

00080 " 8796260

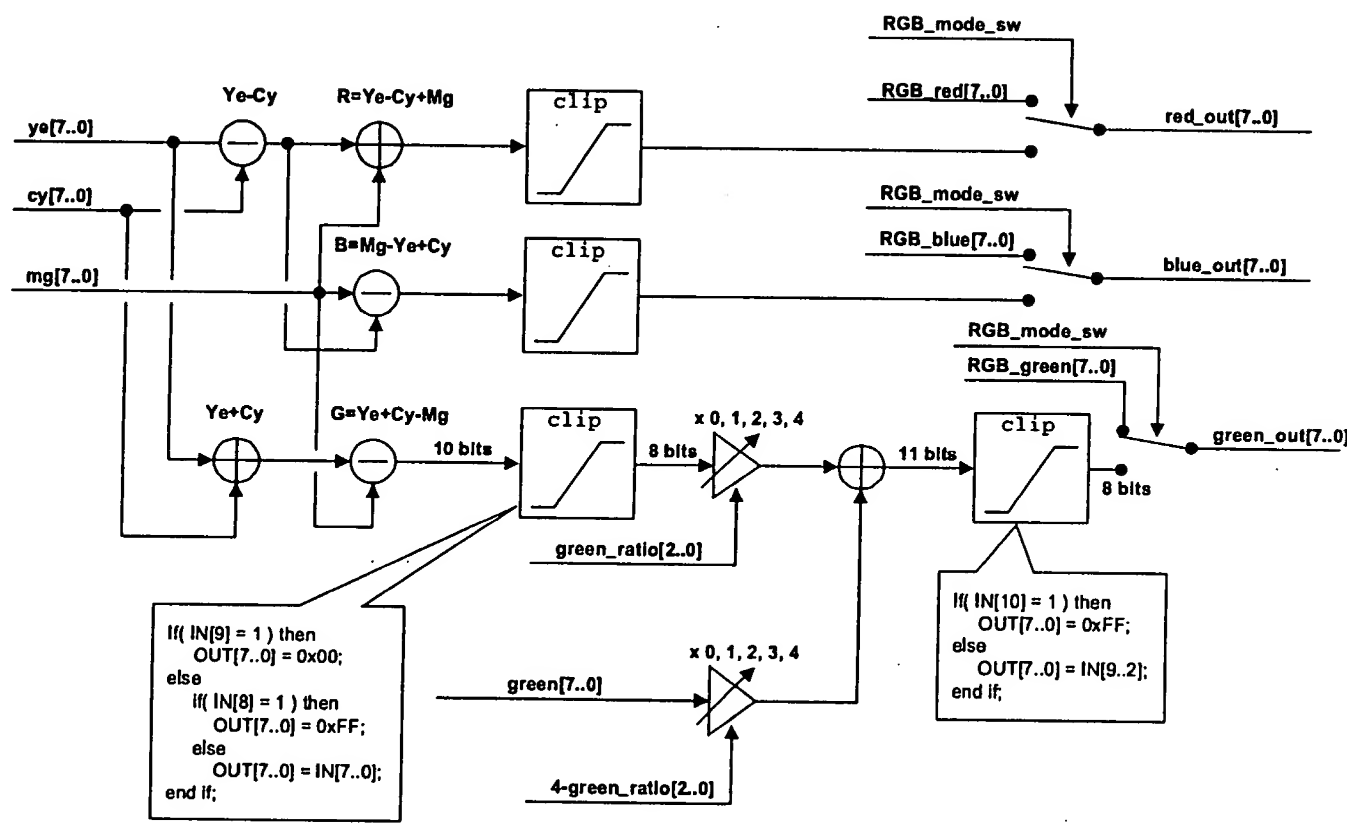


Fig ~~12A~~ 10l Block diagram of comp2RGB conversion module

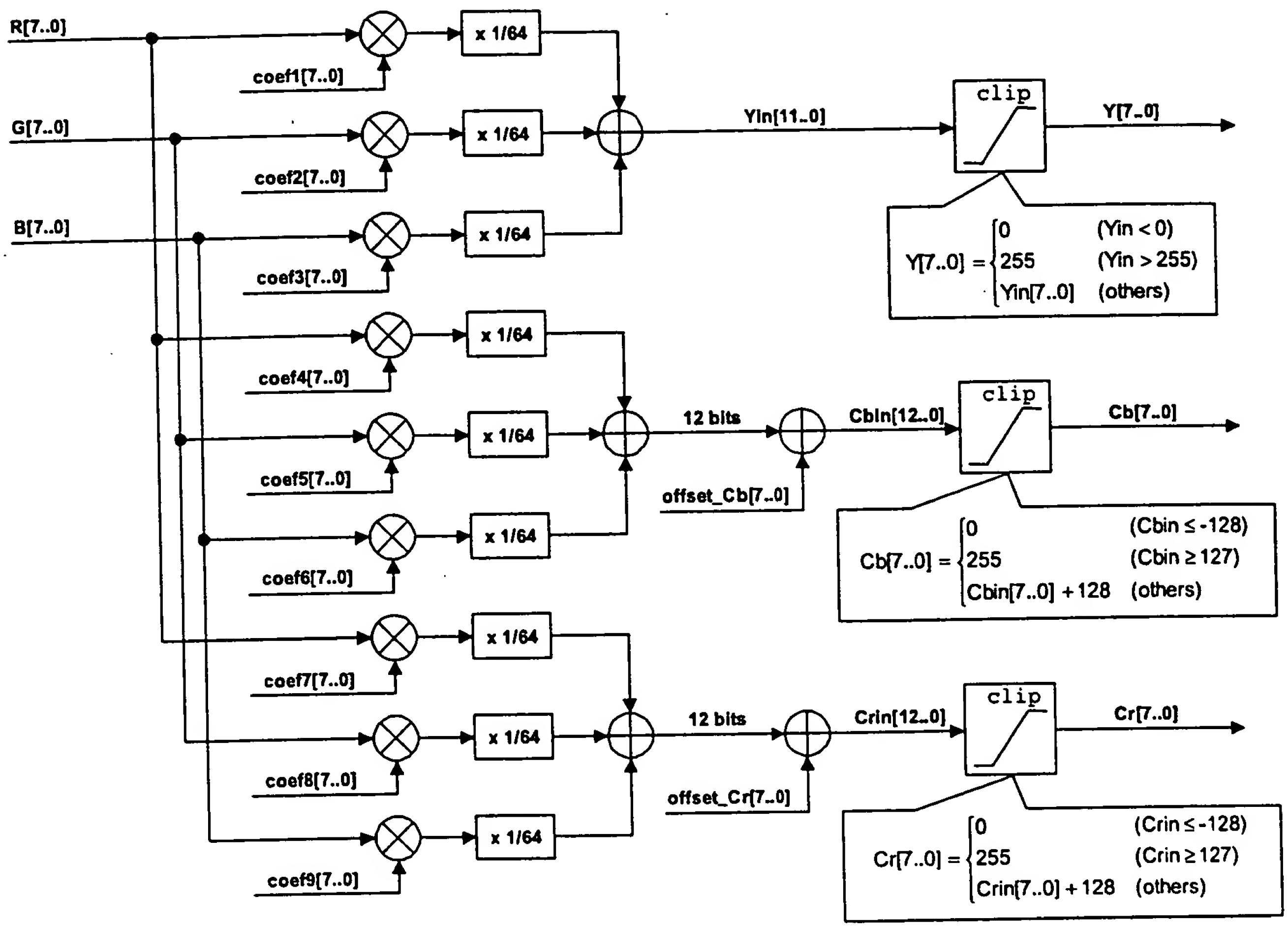
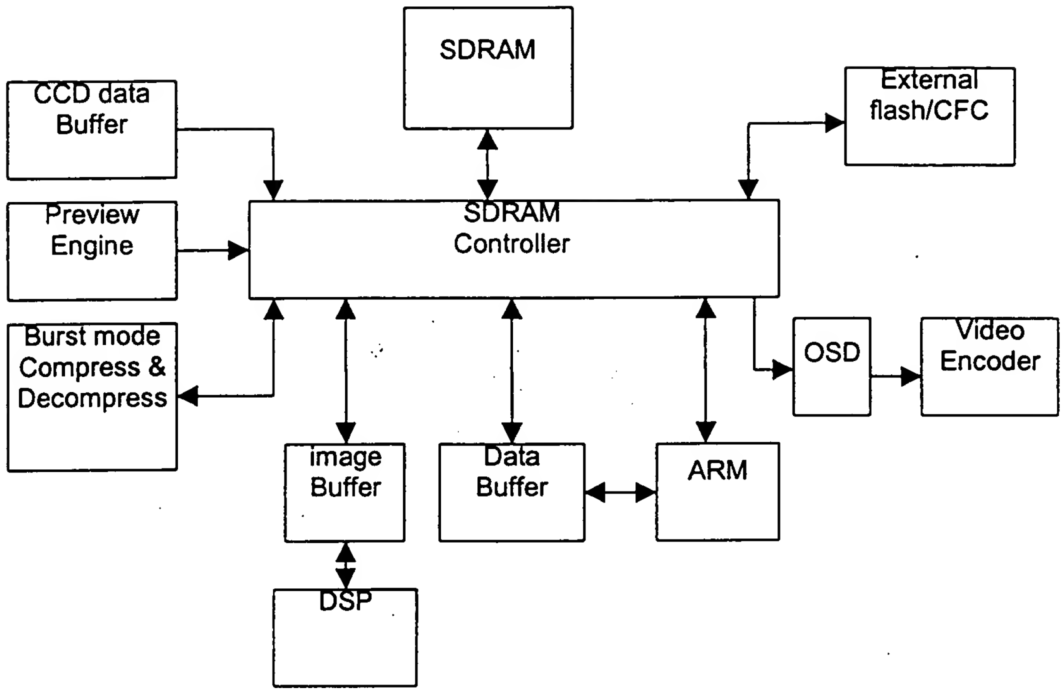


Fig 11 **Block diagram of RGB2YCbCr conversion module**



12a
Figure ~~8~~ SDRAM Controller Data Flow

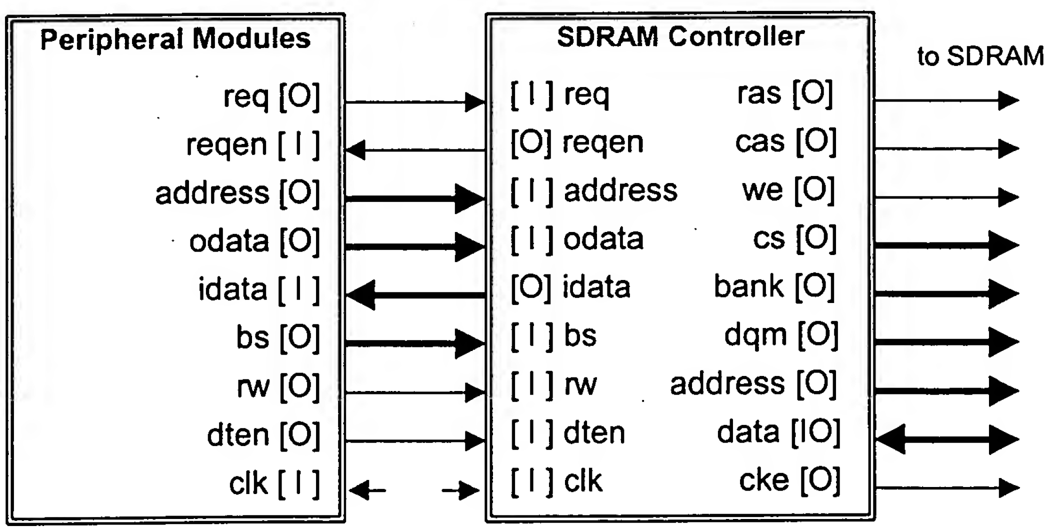


Figure ~~9~~ SDRAM Controller to Peripheral Connections
12b

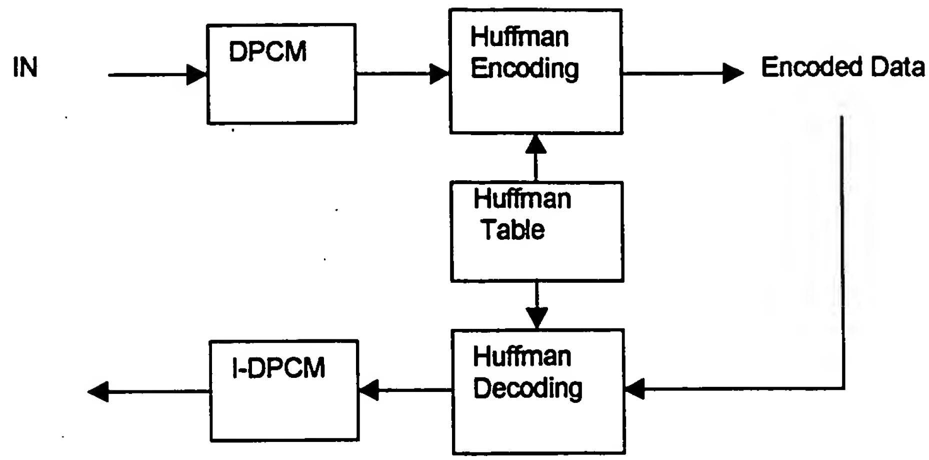
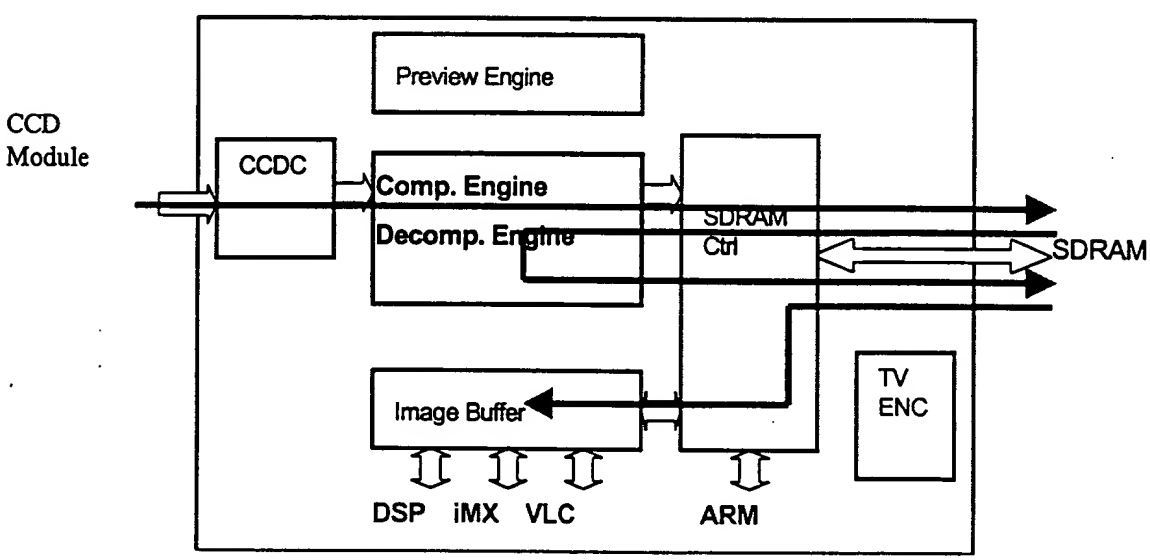


Fig. 13b compression/decompression



• Figure 13a Data Flow in the Burst Mode Compression

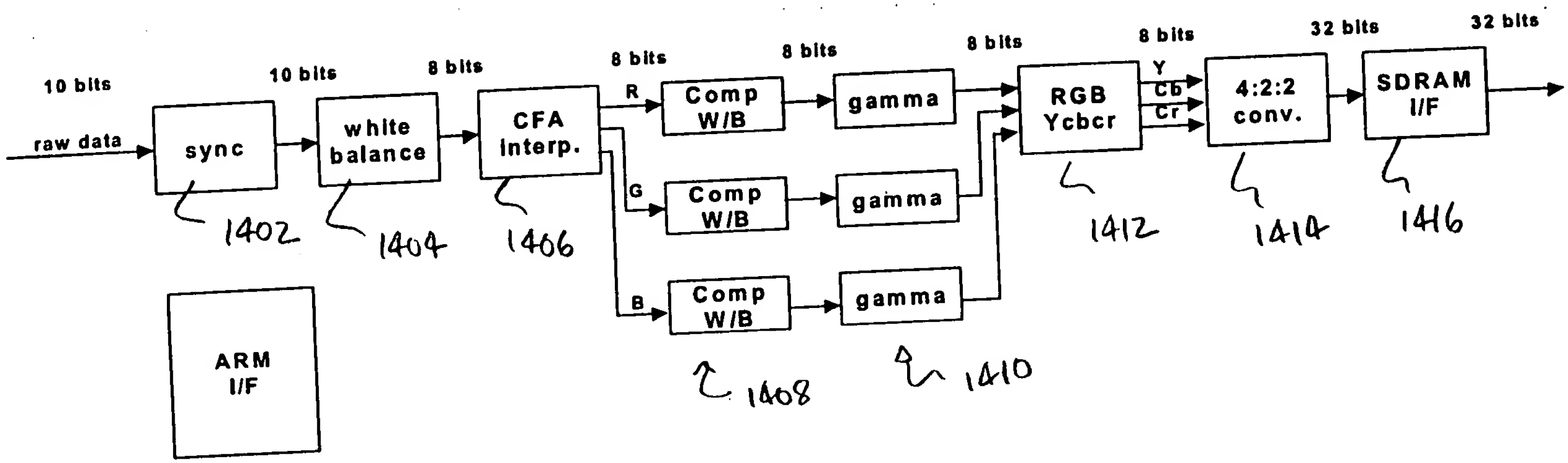


Fig.14 Block diagram of Preview Engine ~~for ES2~~

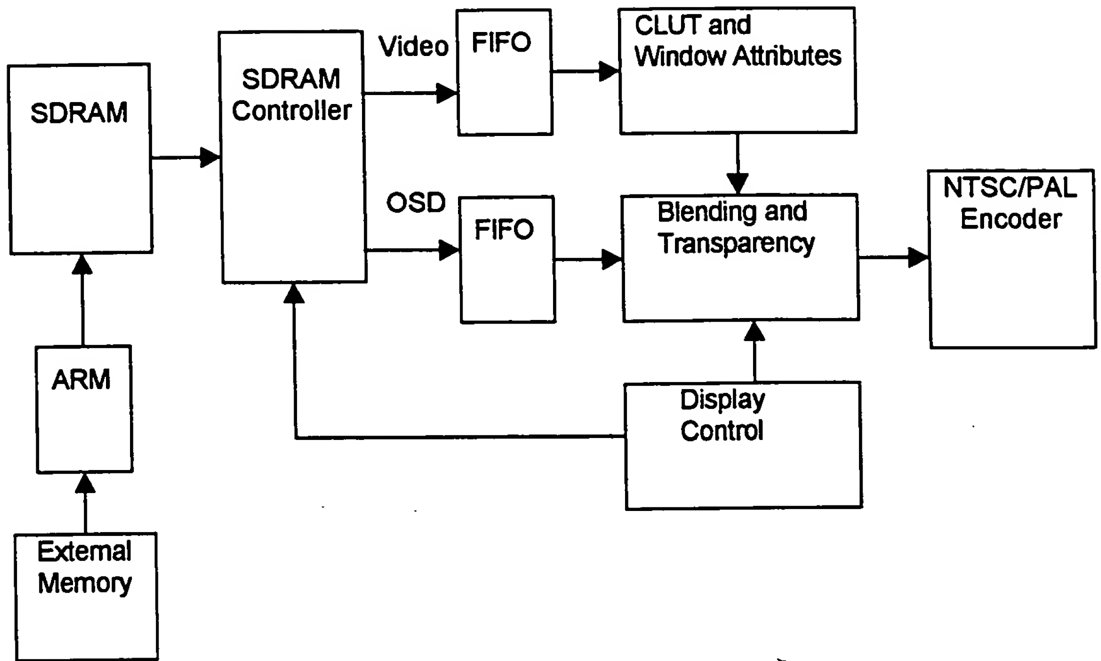

SDRAM
SDRAM Controller
Video
FIFO
CLUT and Window Attributes
OSD
FIFO
Blending and Transparency
NTSC/PAL Encoder
ARM
Display Control
External Memory

Fig. 15 OSD

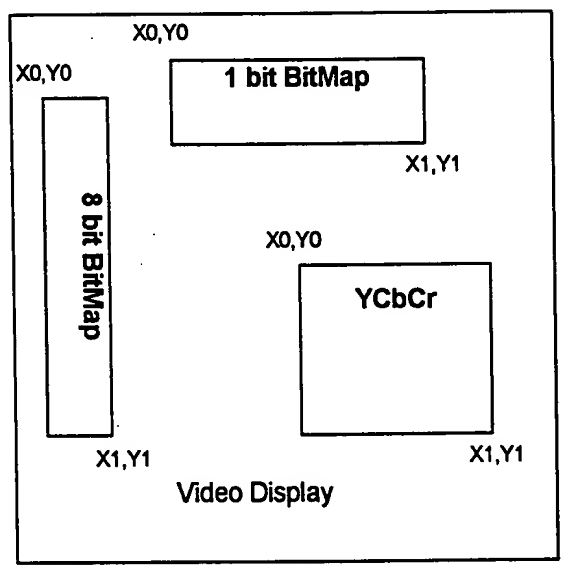

X0,Y0
X0,Y0
1 bit BitMap
X1,Y1
8 bit BitMap
X0,Y0
YCbCr
X1,Y1
X1,Y1
Video Display

Fig. 16

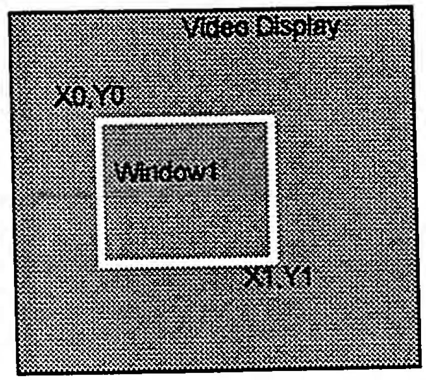
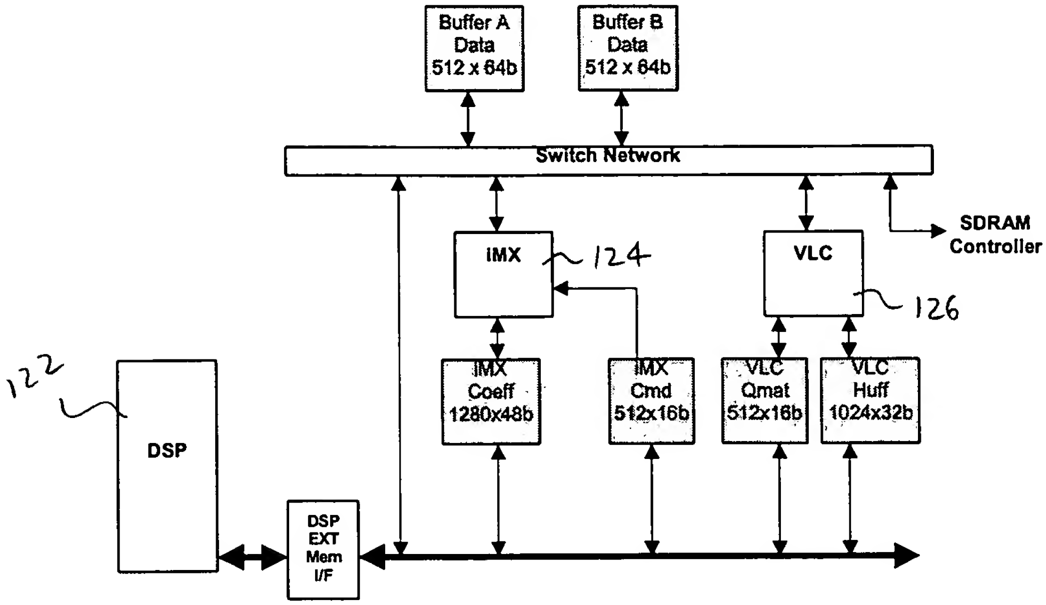
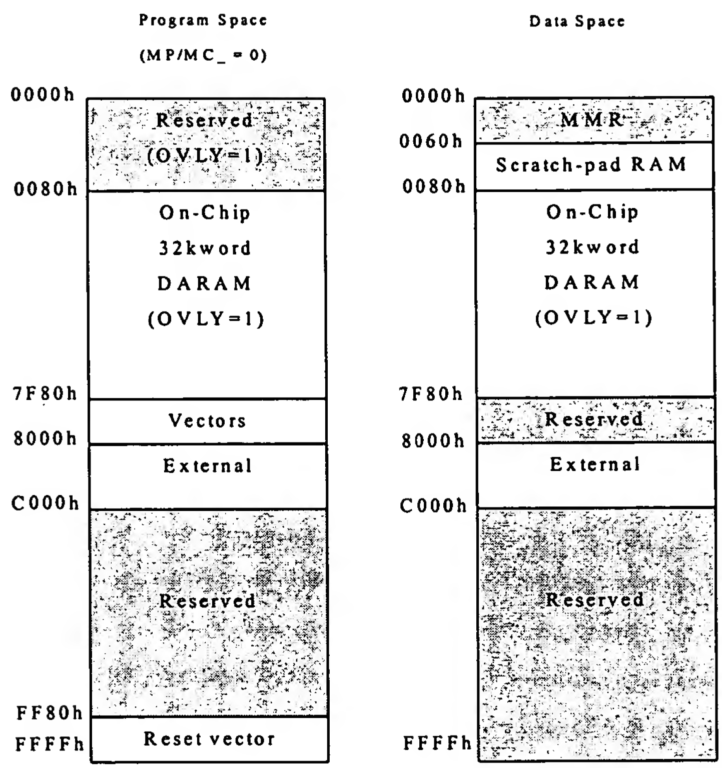


Fig.17



• Figure 18a DSP subsystem block diagram



• Figure DSP Memory Map

18b

TI-28919

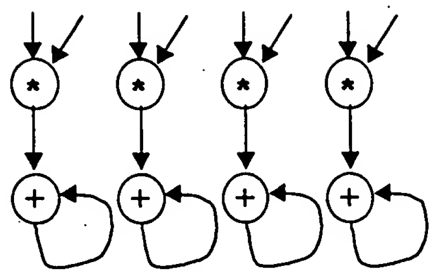
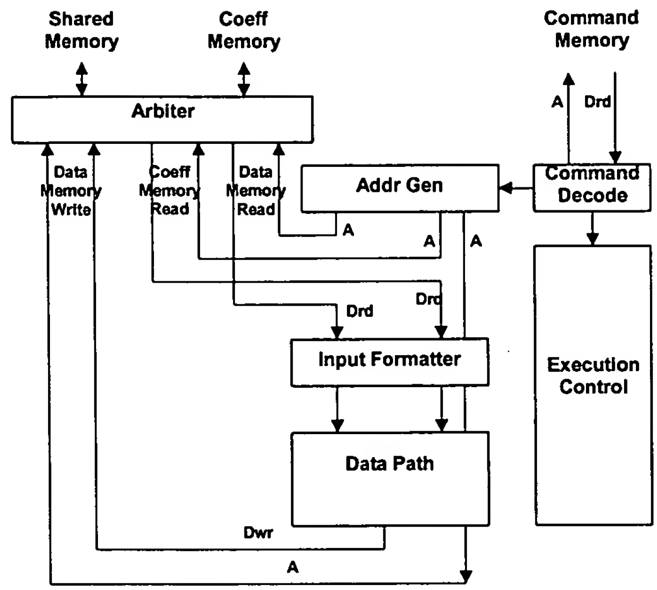
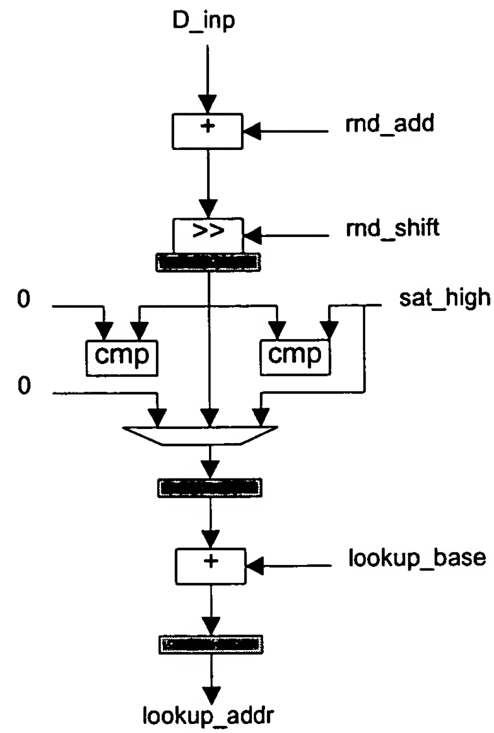


Fig.19



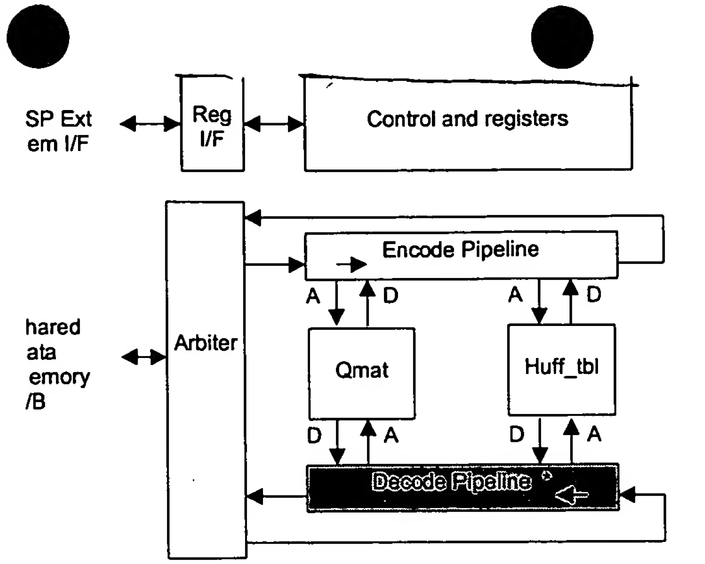
• Figure ~~27~~ 7 iMX Architecture

20



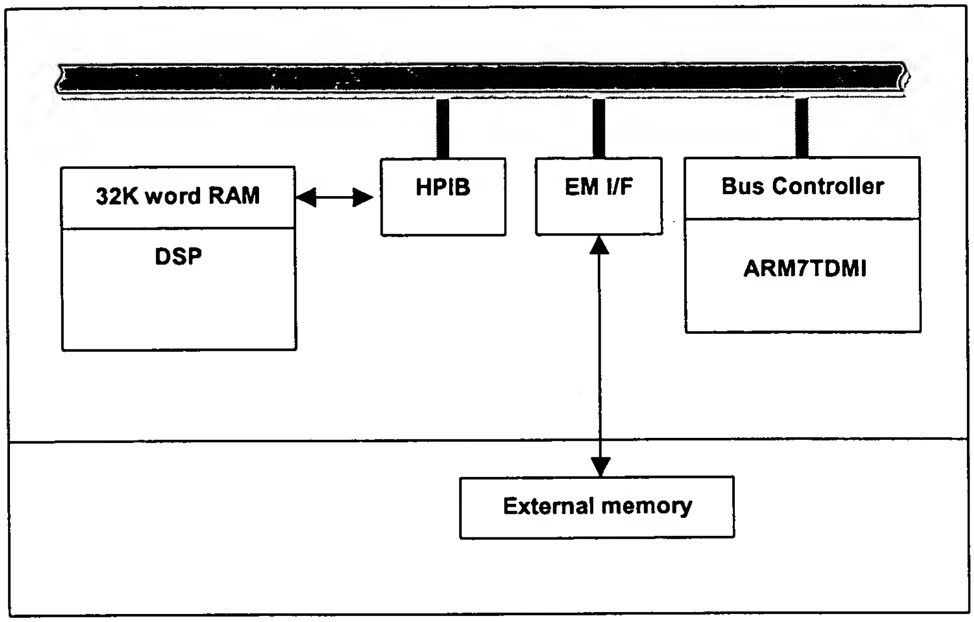
• Figure ~~28~~ Datapath pipeline of table lookup accelerator

21

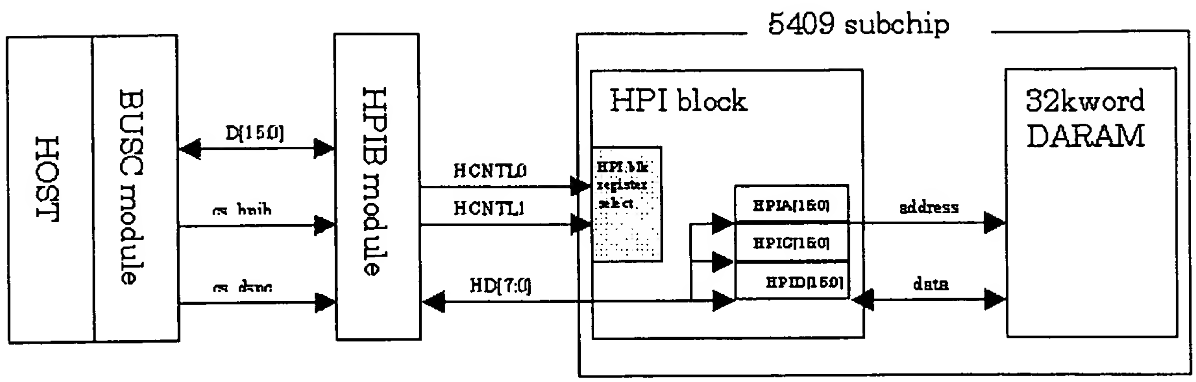


(* decode path not currently implemented)

Fig. 22 VLC Engine block diagram



• Figure 23a Boot phase for DSP and ARM



• Figure DSP and ARM Communication

23b

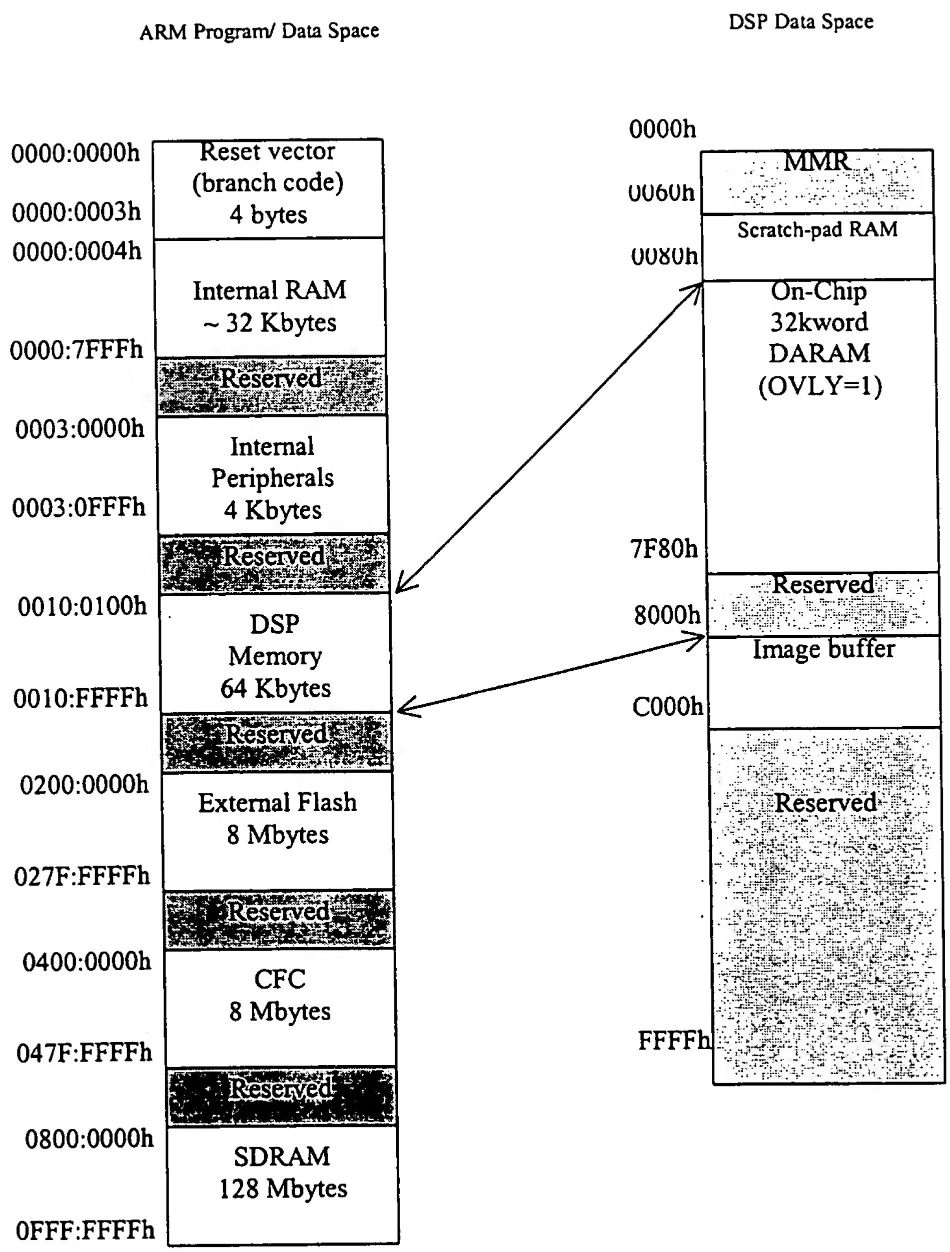


Figure ~~28~~ ARM memory map and relation to DSP memory map

23c

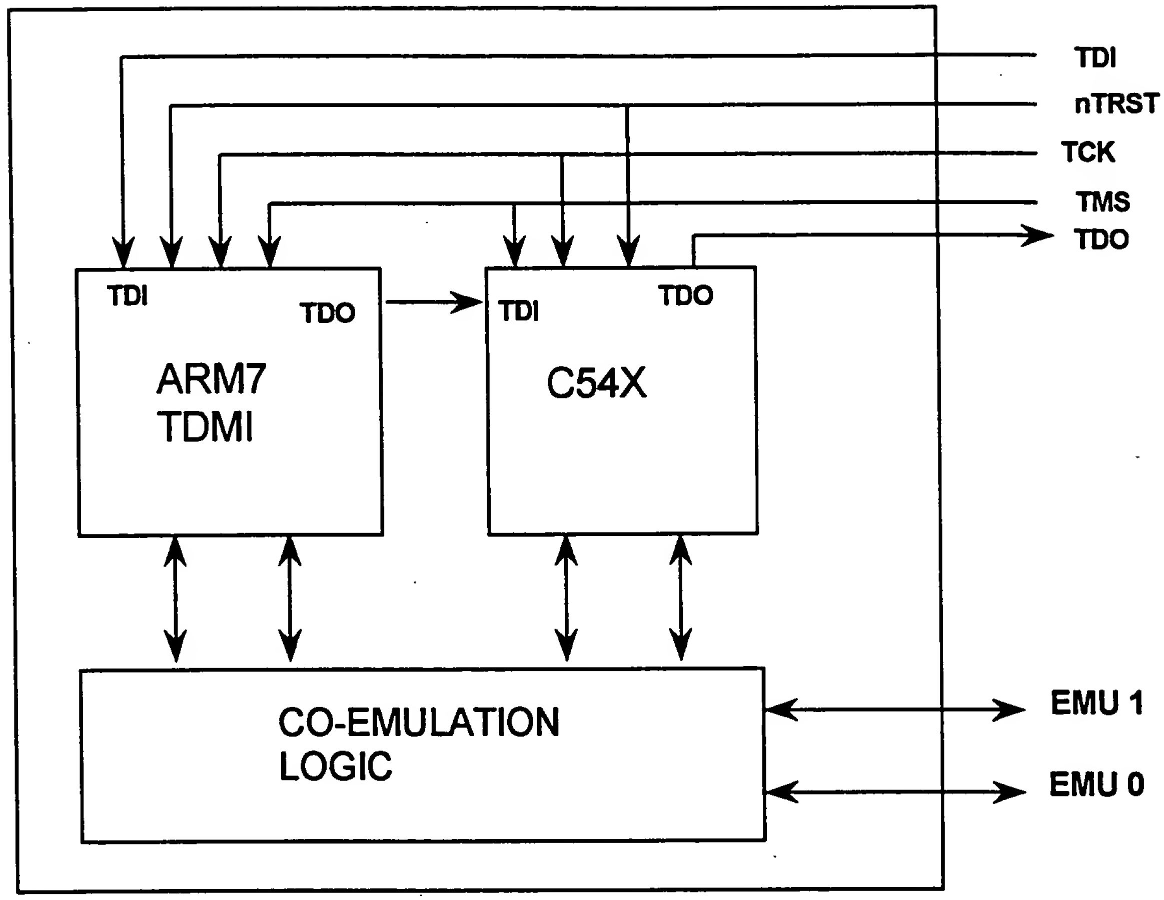

TDI
nTRST
TCK
TMS
TDO
TDI
TDO
ARM7
TDMI
TDI
TDO
C54X
CO-EMULATION
LOGIC
EMU 1
EMU 0

Fig. 24

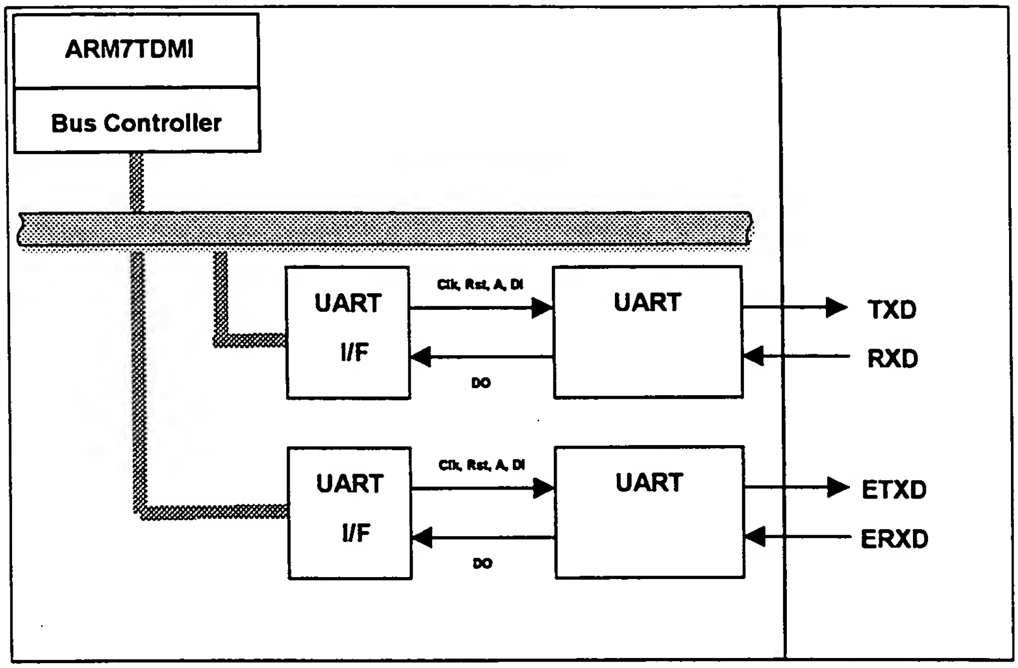

ARM7TDMI
Bus Controller
UART
I/F
Clk, Rst, A, DI
DO
UART
TXD
RXD
UART
I/F
Clk, Rst, A, DI
DO
UART
ETXD
ERXD

Fig. 25

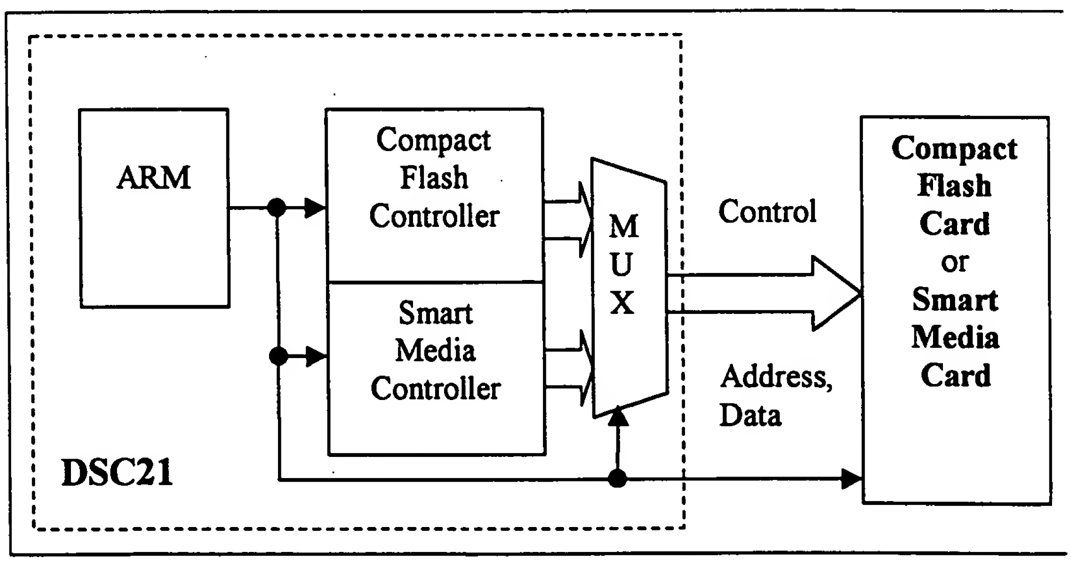

ARM
Compact Flash Controller
Smart Media Controller
M U X
Control
Address, Data
Compact Flash Card or Smart Media Card
DSC21

Fig. 26

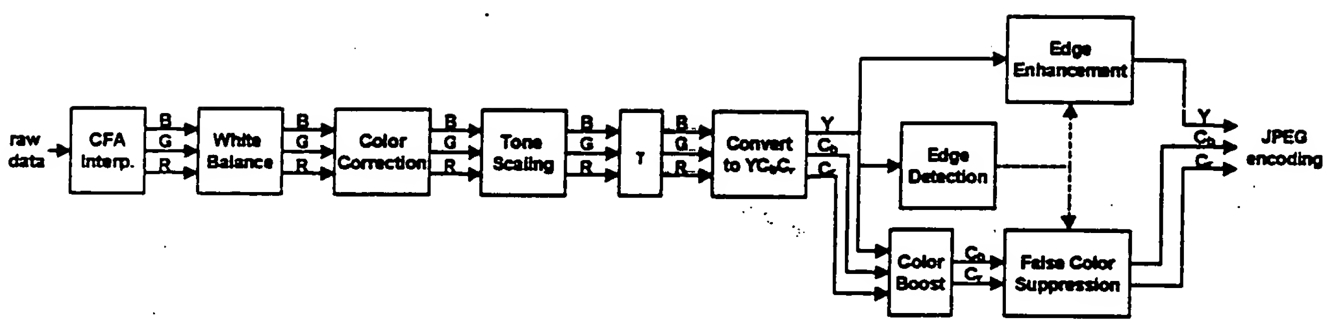
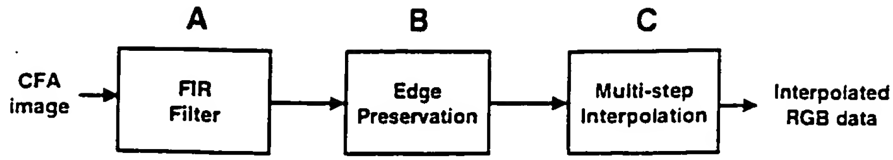


Figure 1. DSC Preprocessing Block Diagram

27



~~Fig 1~~ 2

Fig 1 Block diagram of the CFA multi-step interpolation with edge preservation

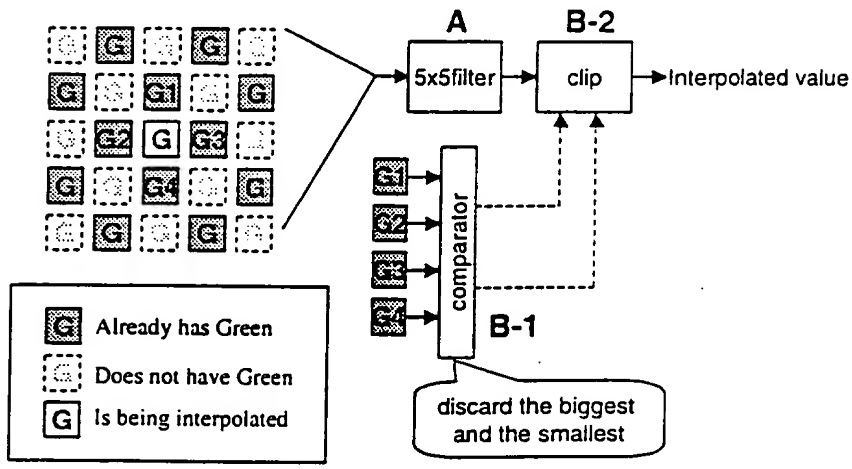


Fig 2 Edge preservation filtering for green

29

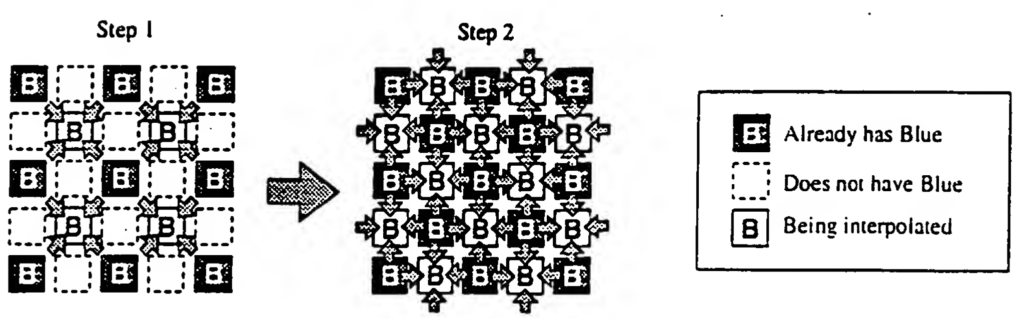


Fig 3 Two-step interpolation for blue and red

28

R	G	R	G
G	B	G	B
R	G	R	G
G	B	G	B

Figure-1 RGB Bayer pattern

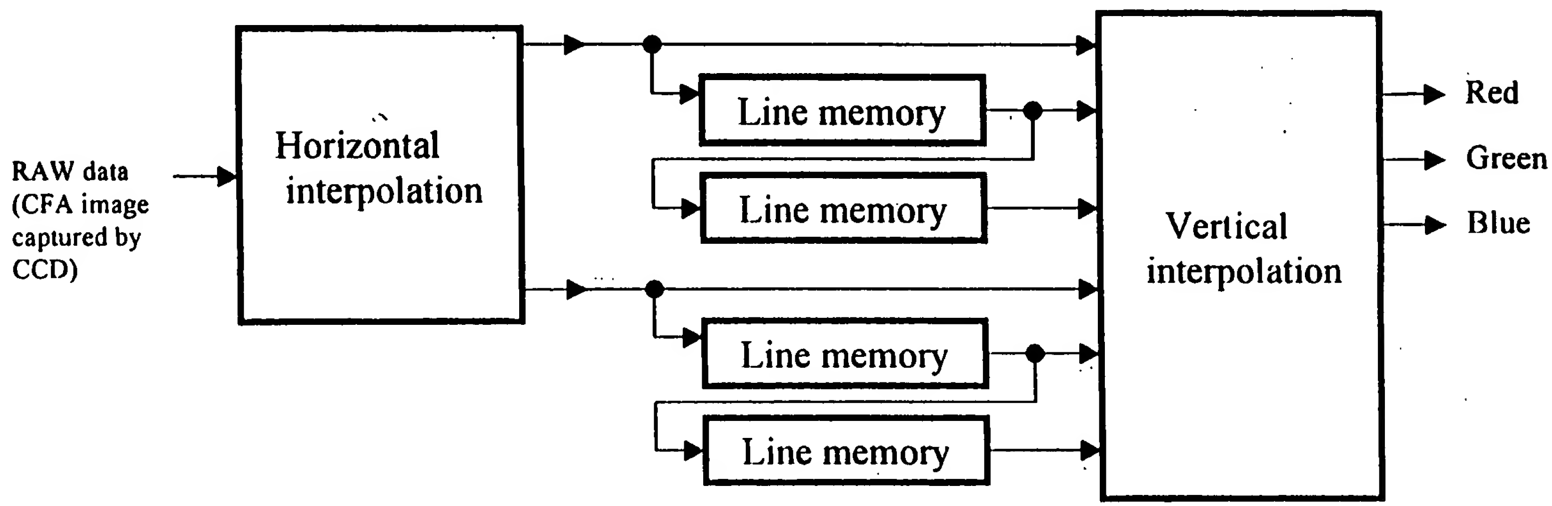
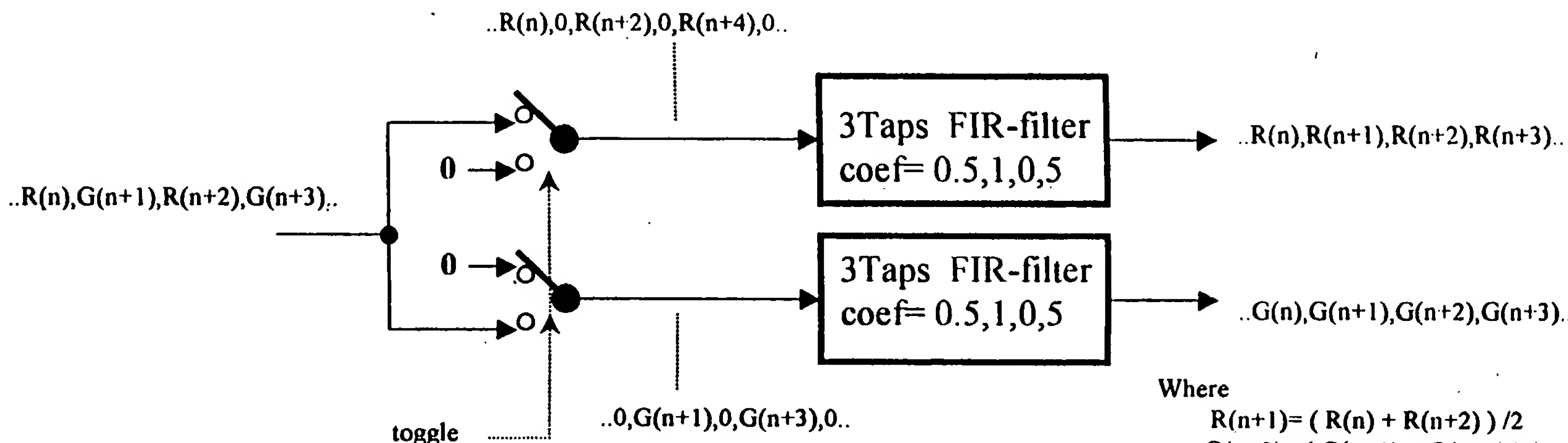
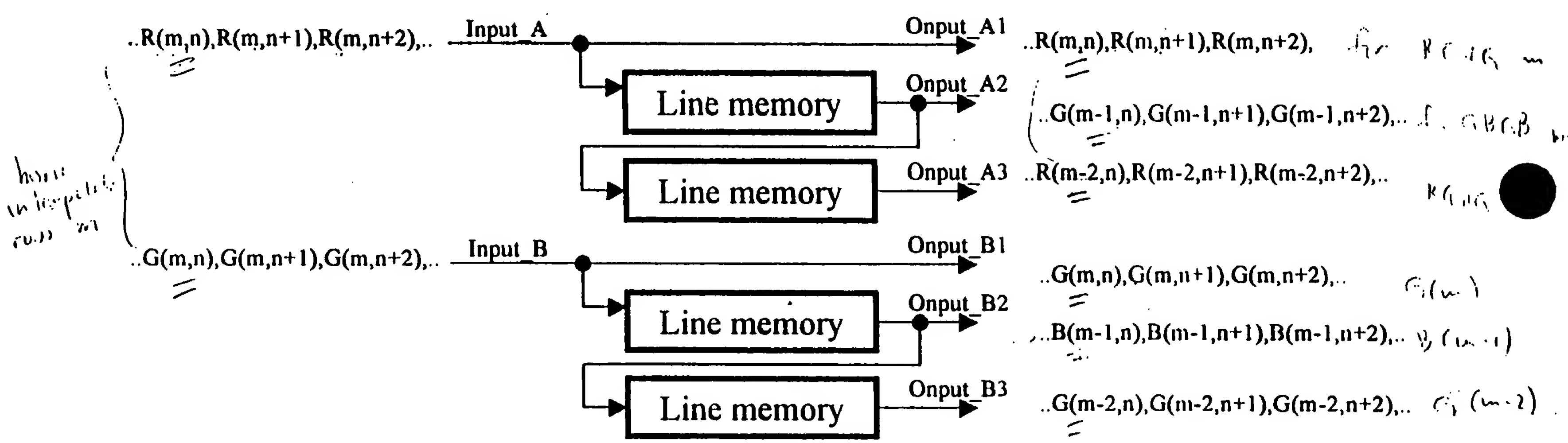


Figure-2 A Block diagram of prior CFA interpolation



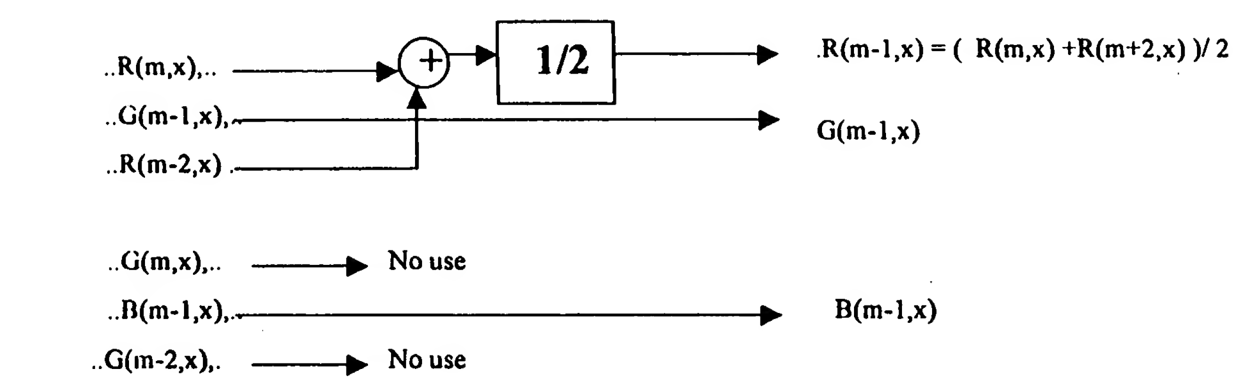
Where
R(n+1)= (R(n) + R(n+2)) /2
G(n+2)= (G(n+1) + G(n+4)) /2

Figure-3 A Block diagram of Horizontal interpolation
31

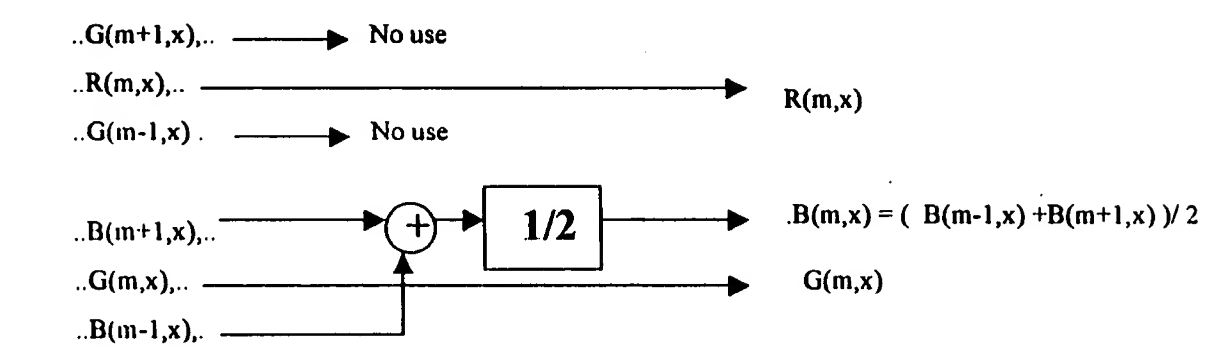


Where
G(y,x) is a data at the position
, "vertical= y / horizontal= x ".

Figure-4 Four line memories
32



(a) **Odd line (m-1) , m=2*y**



(b) **Even line (m) , m=2*y**

Figure-~~5~~ 7 A Block Diagram of Conventional Vertical Interpolation

33

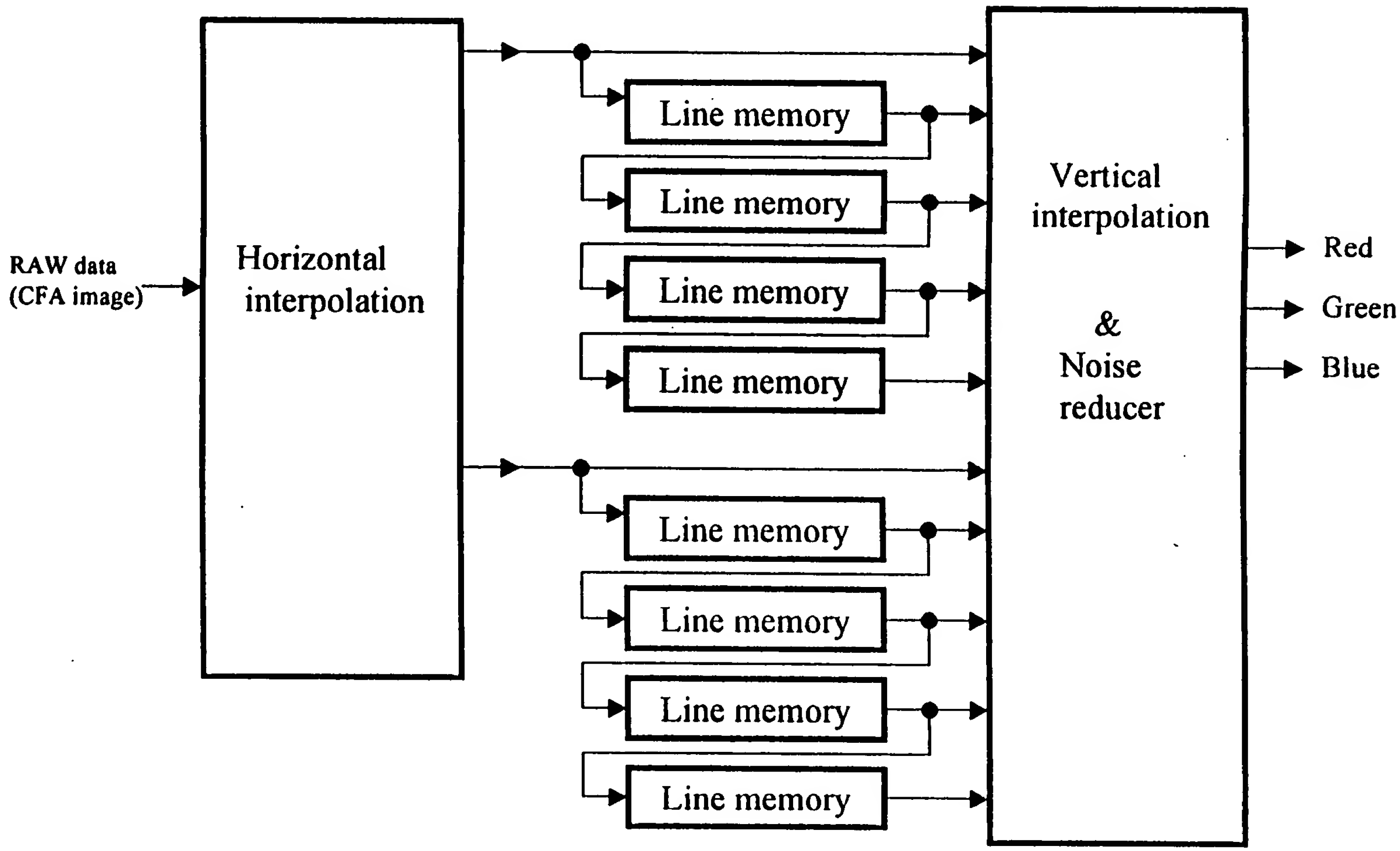
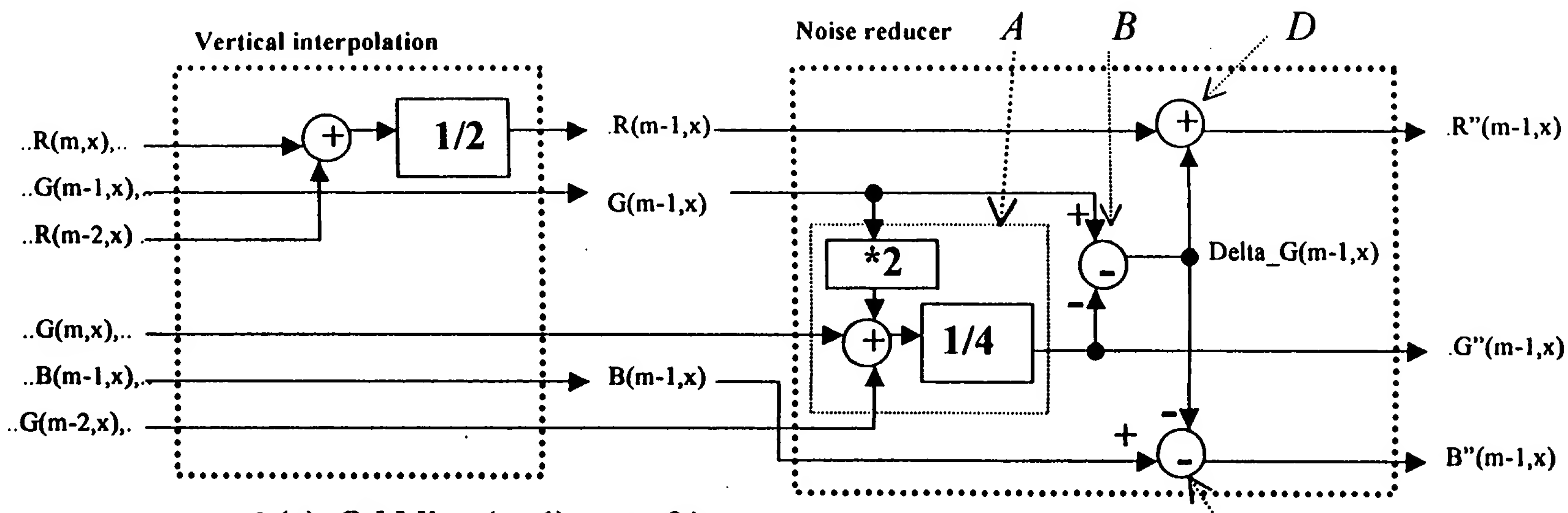


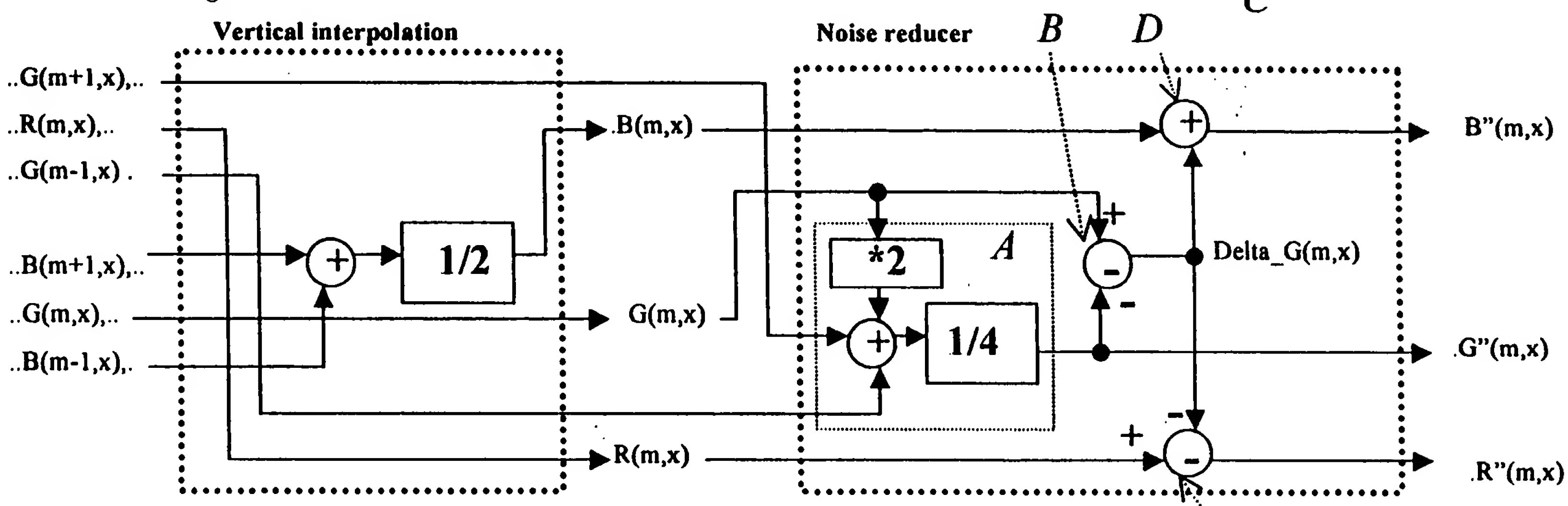
Figure-~~6~~ A Block diagram of conventional CFA interpolation with Noise reducer

34

00400080" 09652543



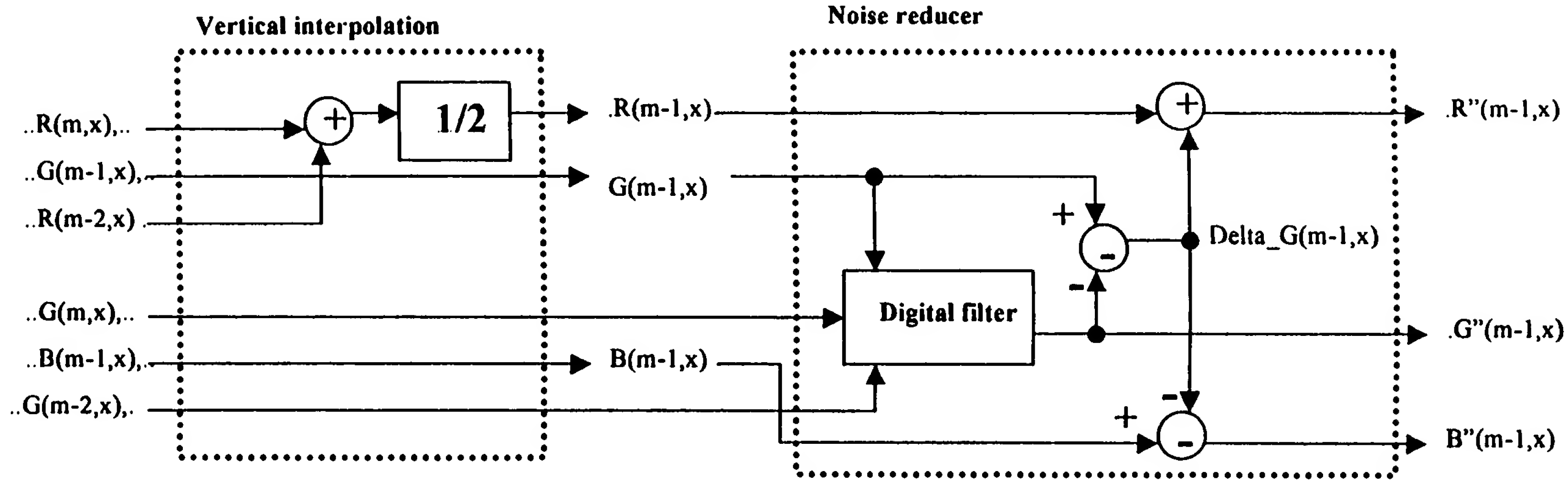
OUT (a) Odd line (m-1) , m=2*y



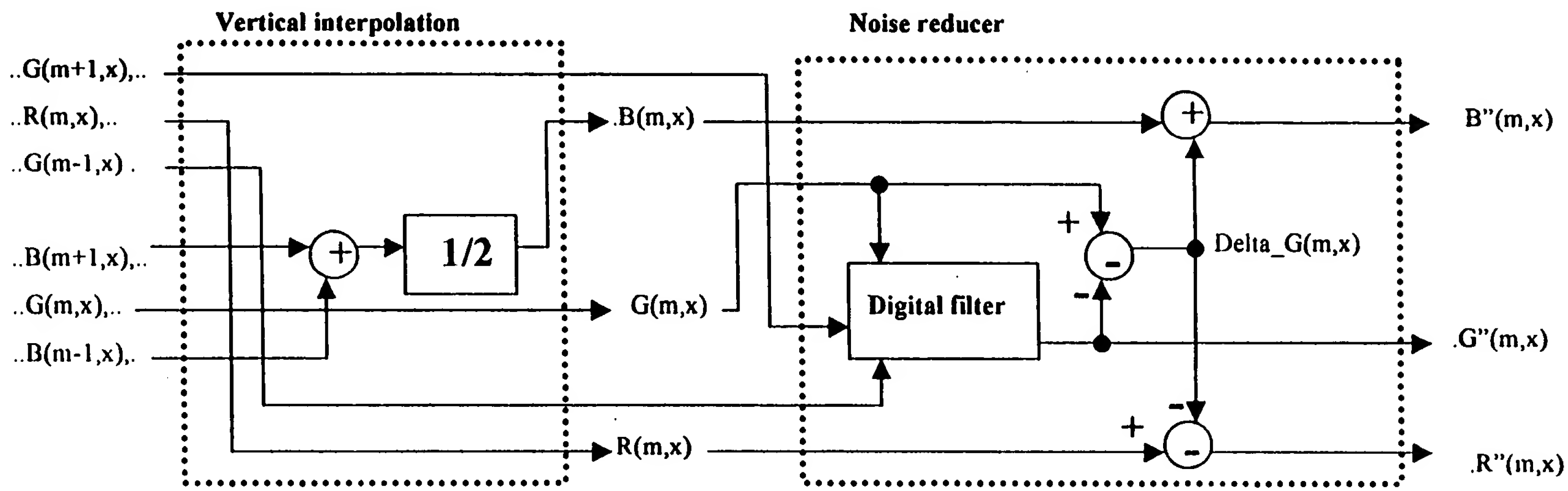
OUT (b) Even line (m) , m=2*y

Figure-7 A Vertical interpolation and Noise reducer

35



(a) Odd line (m-1) , m=2*y

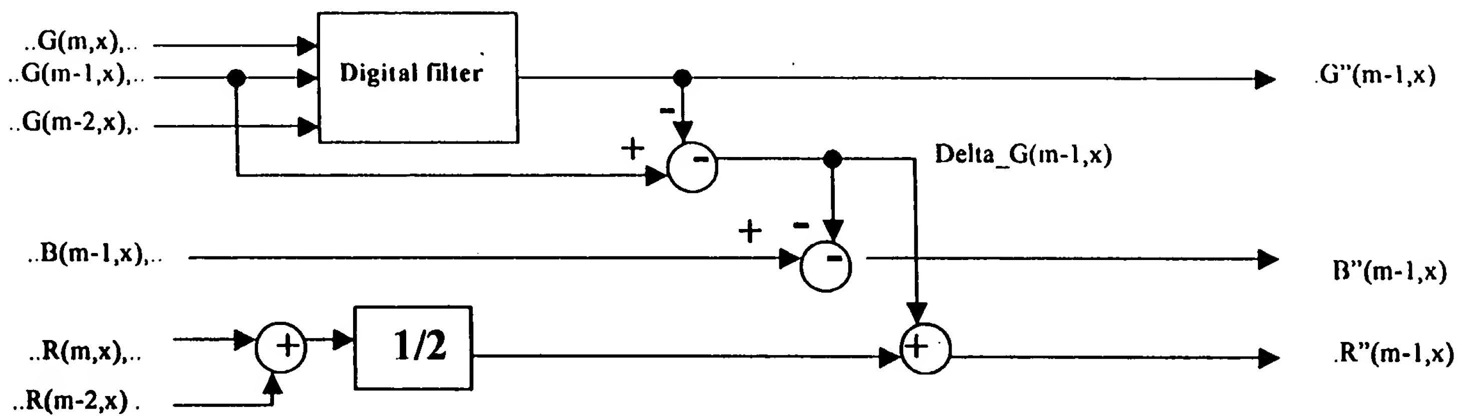


(b) Even line (m) , m=2*y

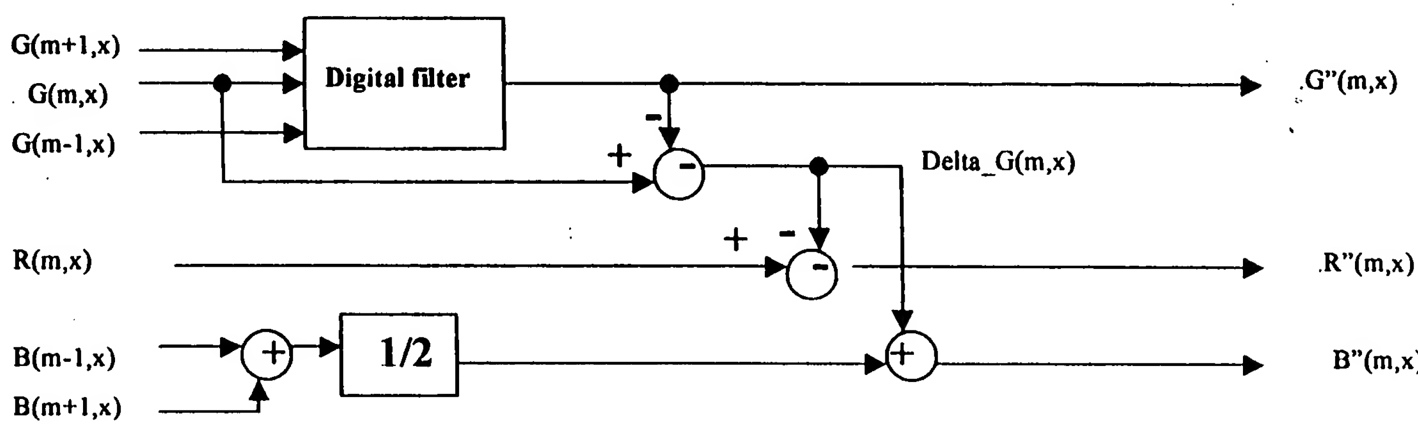
Note: Digital filter is equivalent to "A" shown in Figure-7.

Figure-9 A Vertical interpolation and Noise reducer(Digital filter)

36



(a) Odd line (m-1) , m=2*y



(b) Even line (m) , m=2*y

Figure-10 Another Vertical interpolation with Noise reducer

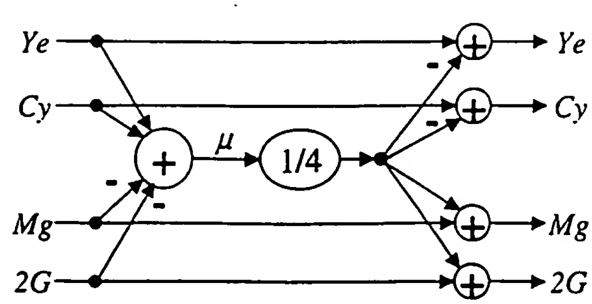
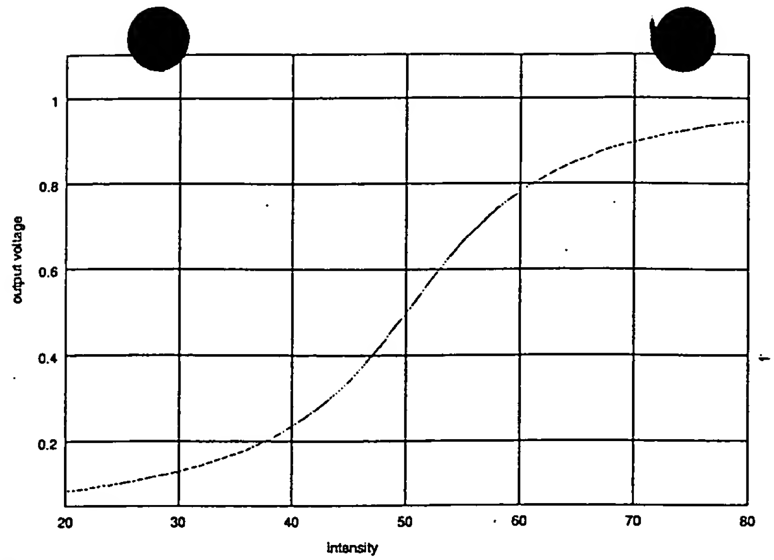


Figure ~~3~~ Block Diagram of Inter-Color Image Enhancement
38



39a

Figure 1: Model for the sensor response output over the input light intensity.

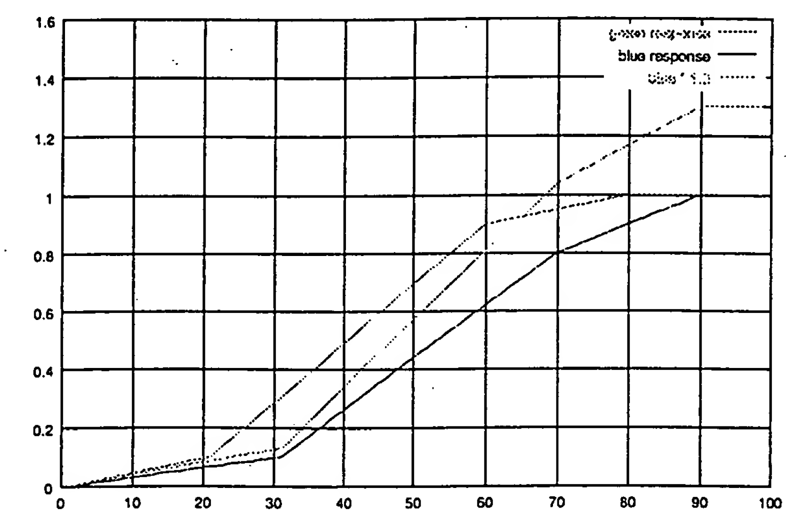
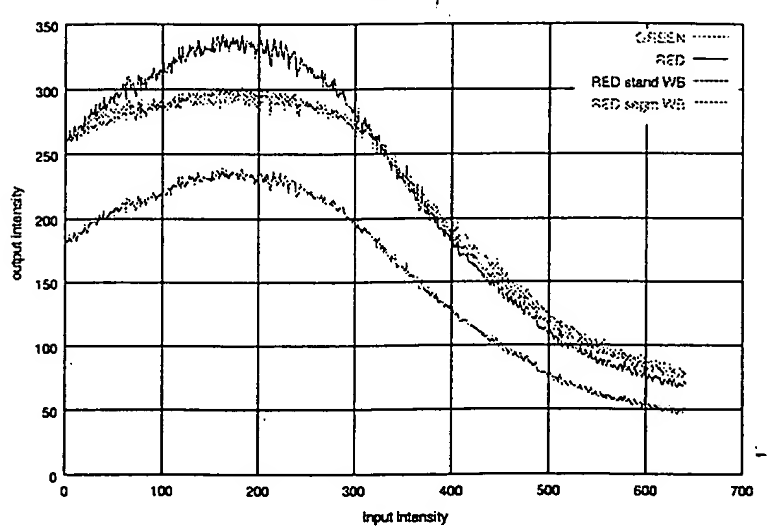


Figure 39b:



40

Figure 3: Example of white balancing the red signal with respect to the green signal: a single multiplier exceeds the green signal in bright areas, whereas the segmented white balancing matches the green curve for all intrensities.

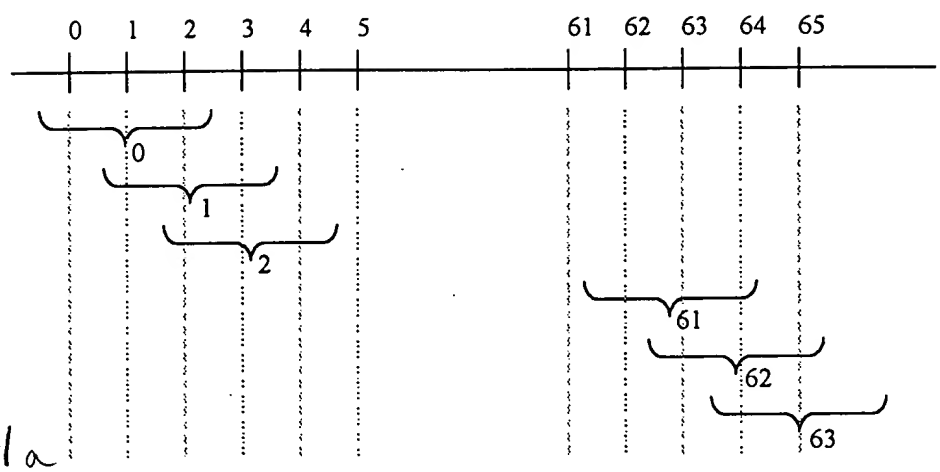


Fig. 41a

Since we need to produce 63 outputs for every 64 inputs, the outputs are 64/63 apart. We can express the output position as

$$outp\_pos[j] = 1 + j*64/63$$

For now the filter kernel is represented as a symmetrical continuous function f(t) centered at time 0. Output[0], for example, needs 3 values: f(-1), f(0), and f(1).

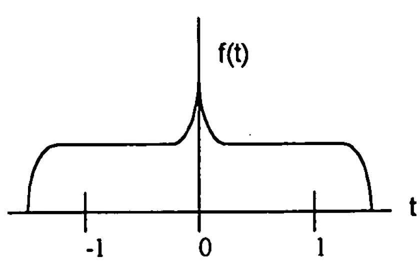


Fig. 41b

Each output point is computed as inner product of 3 coefficient values with 3 input points.

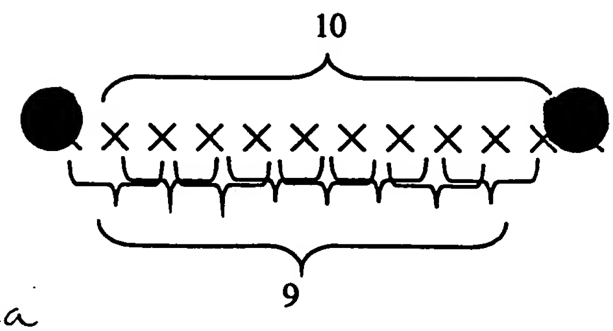


Fig. 42a

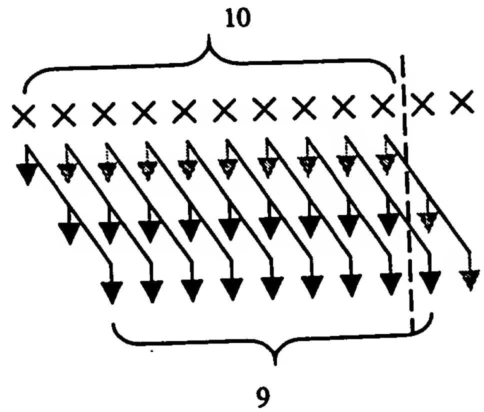


Fig. 42b

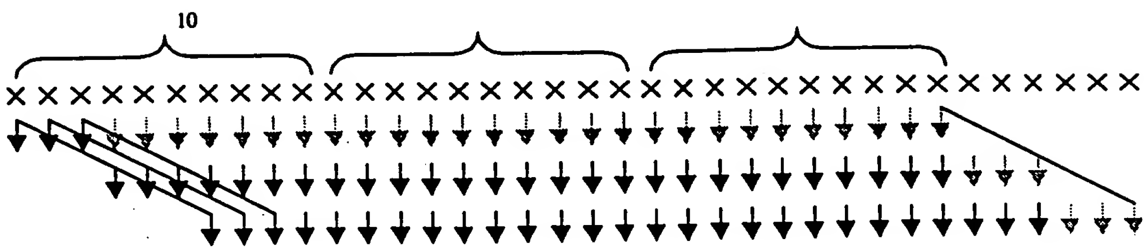


Fig. 42c

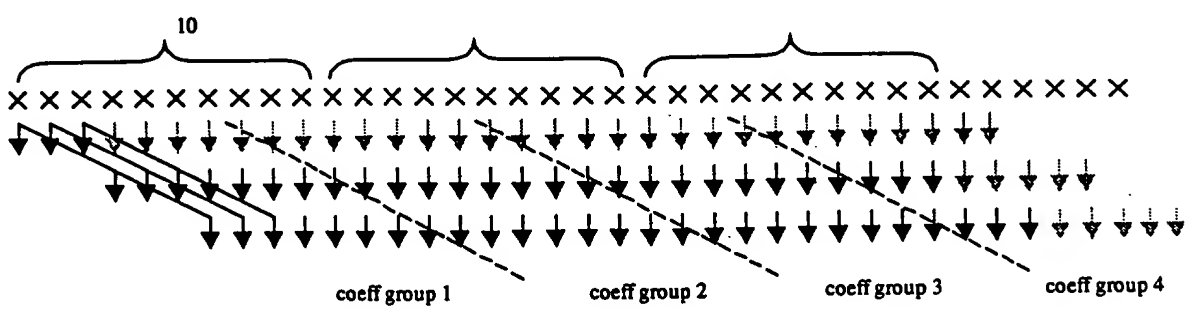


Fig. 42d

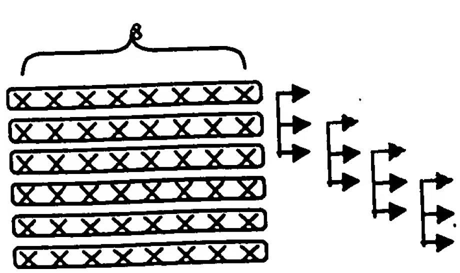


Fig. 42e

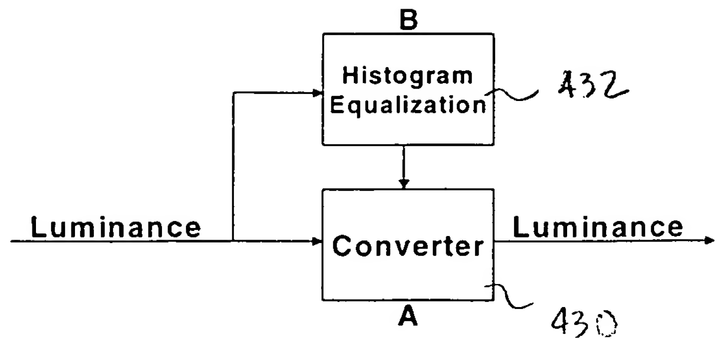
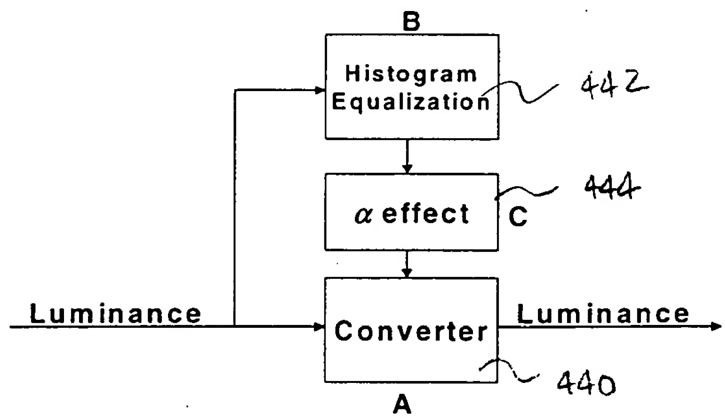


Figure 1. Block Diagram of the former example of the Tone-scaling

43



44

Figure 2. Block Diagram of the example of the Invented Tone-scaling

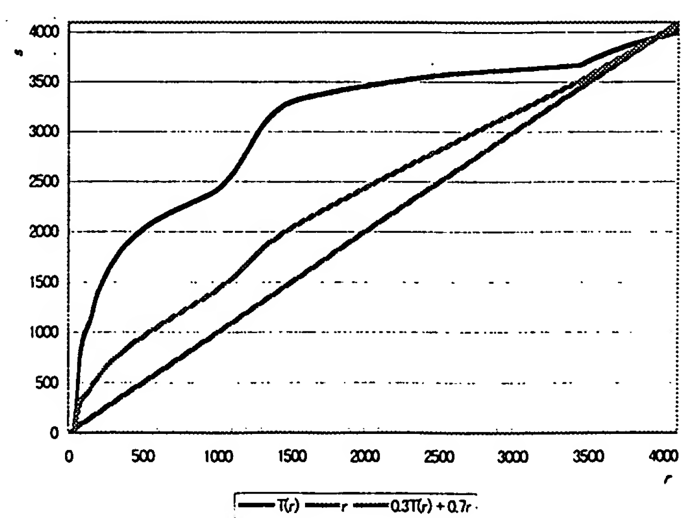


Figure 3. Curves of the Tone-scalings

45

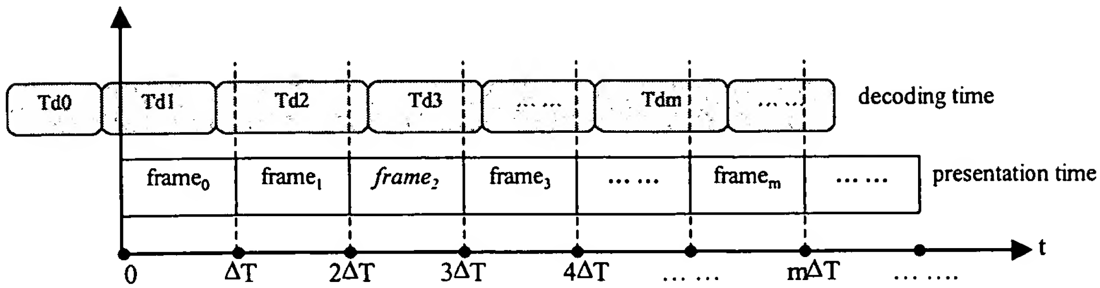


Fig. 46 (a) Synchronization is lost after frame 1

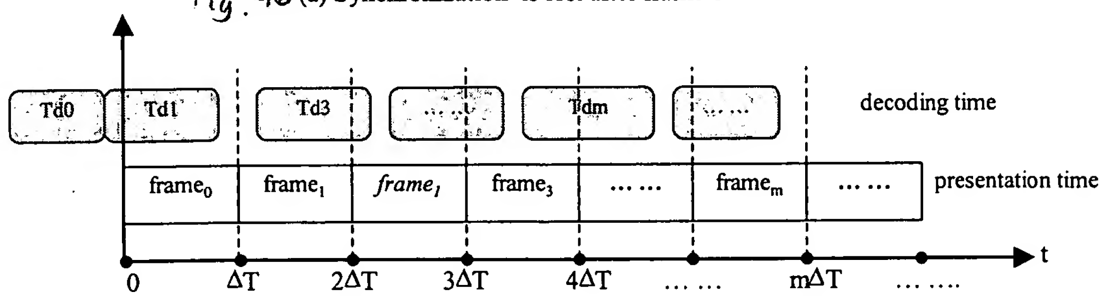


Fig. 46 (b) Synchronization is resumed at frame 3 by skipping frame 2

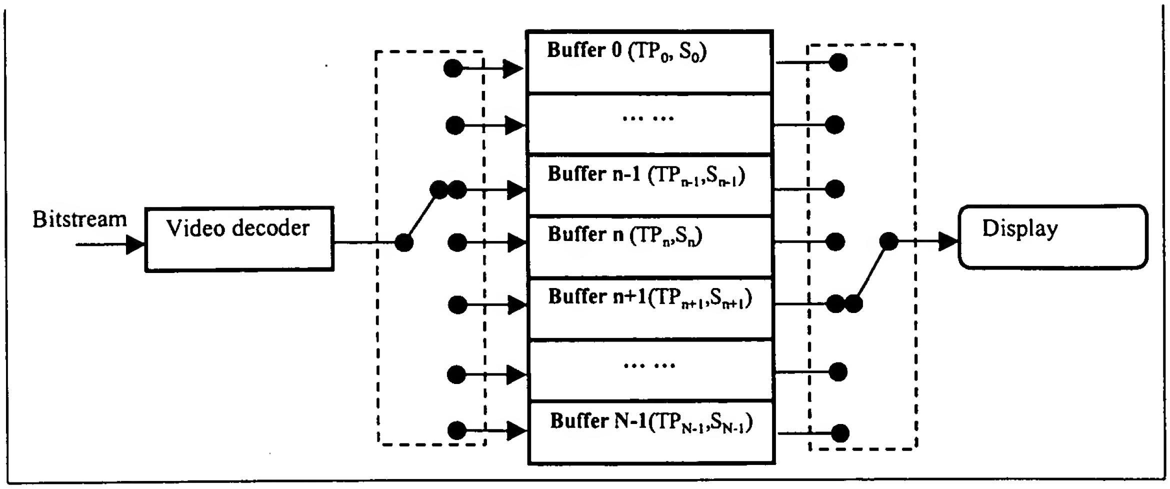
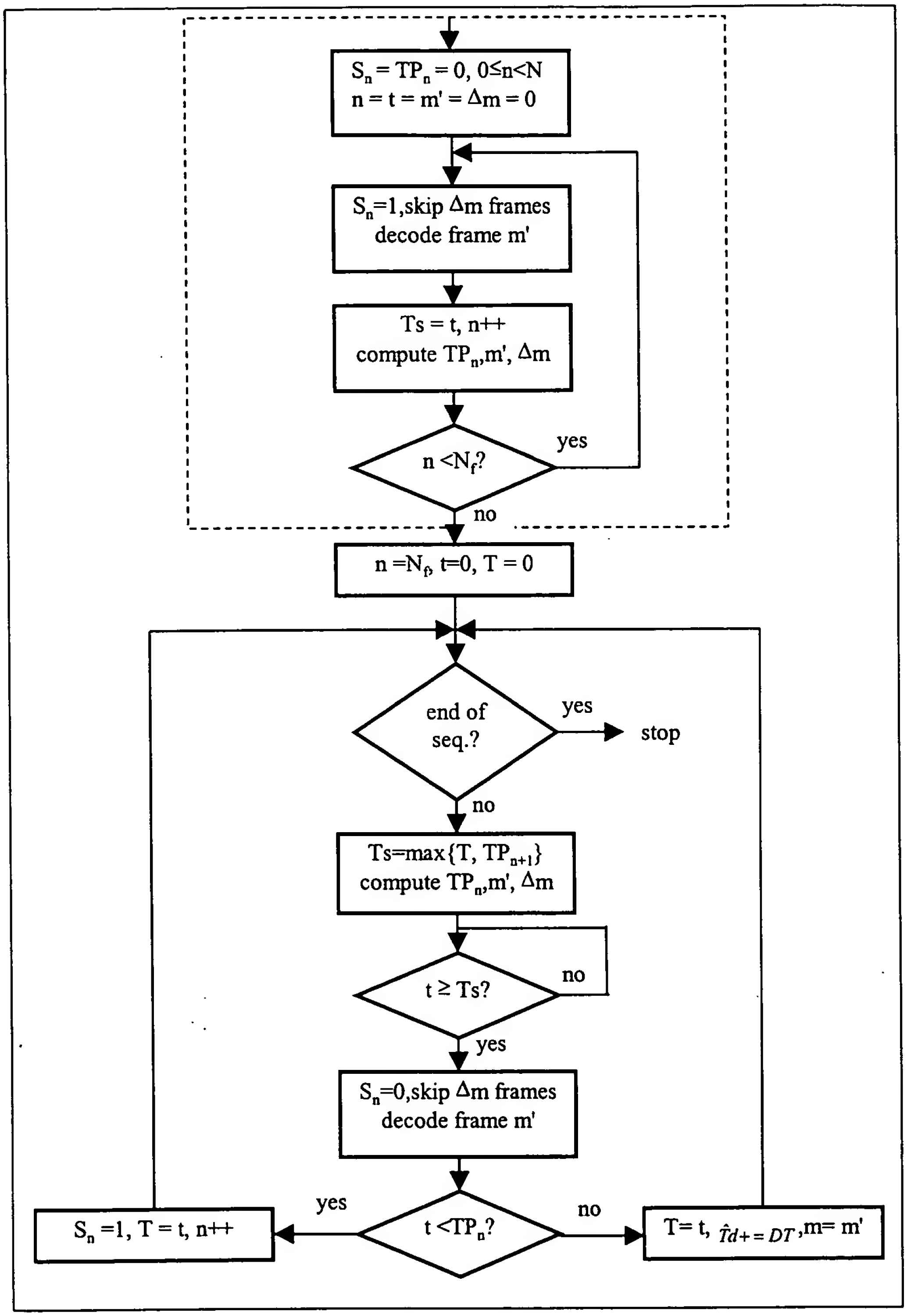

Bitstream
Video decoder
Buffer 0 (TP0, S0)
... ...
Buffer n-1 (TPn-1,Sn-1)
Buffer n (TPn,Sn)
Buffer n+1(TPn+1,Sn+1)
... ...
Buffer N-1(TPN-1,SN-1)
Display

Fig. 47



$S_n = TP_n = 0, 0 \le n < N$
$n = t = m' = \Delta m = 0$
S_n=1,skip Δm frames
decode frame m'
Ts = t, n++
compute TP_n,m', Δm
n <N_f?
yes
no
n =N_f, t=0, T = 0
end of seq.?
yes
stop
no
Ts=max{T, TP_{n+1}}
compute TP_n,m', Δm
t ≥ Ts?
no
yes
S_n=0,skip Δm frames
decode frame m'
S_n =1, T = t, n++
yes
t <TP_n?
no
T= t, $\hat{T}d$+= DT,m= m'

Fig. 48

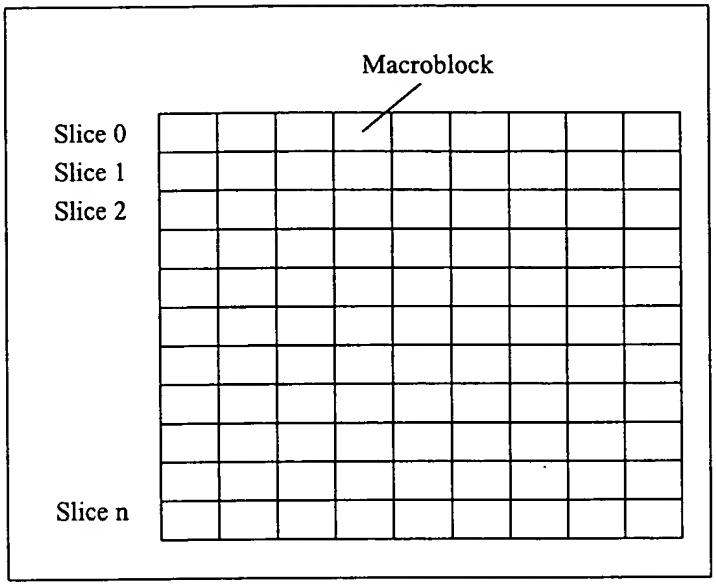


Fig. 49

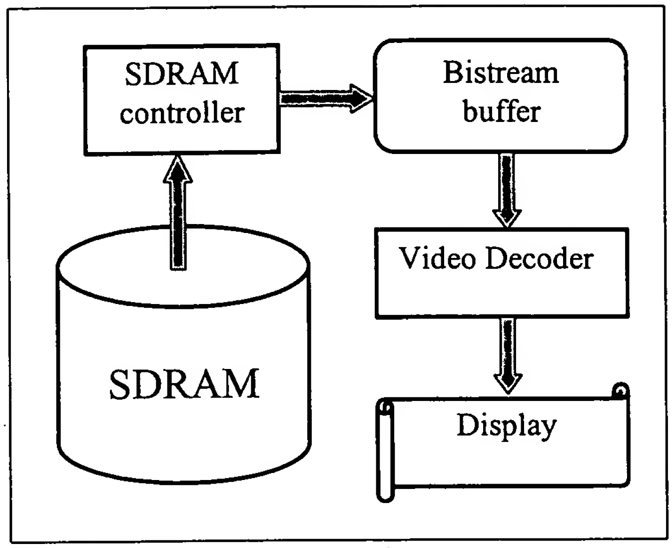

SDRAM
controller
Bistream
buffer
Video Decoder
SDRAM
Display

Fig. 50

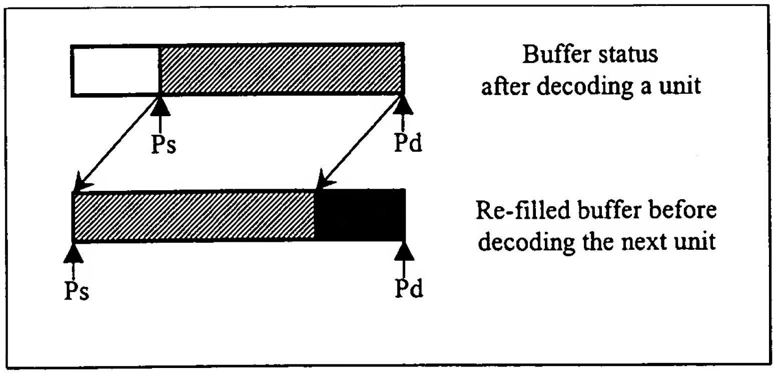

Buffer status
after decoding a unit
Ps
Pd
Re-filled buffer before
decoding the next unit
Ps
Pd

Fig. 51a

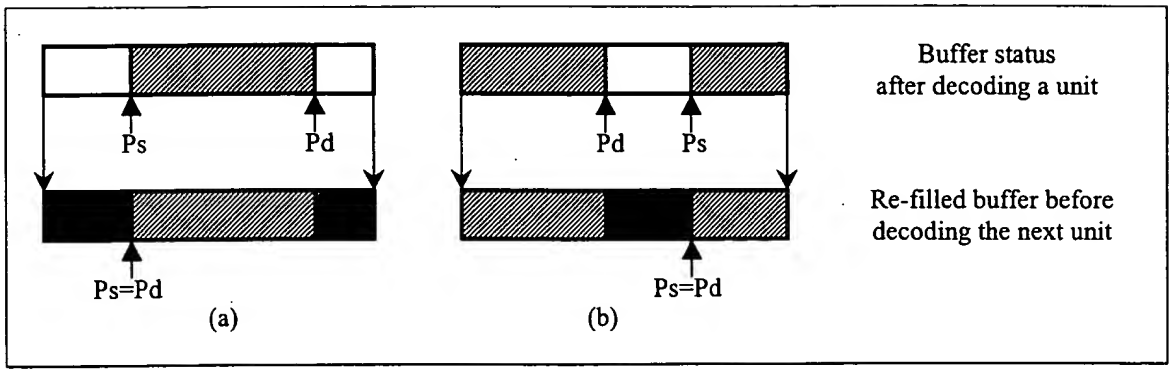

Buffer status
after decoding a unit
Re-filled buffer before
decoding the next unit
Ps
Pd
Ps=Pd
(a)
Pd
Ps
Ps=Pd
(b)

Fig 51 b

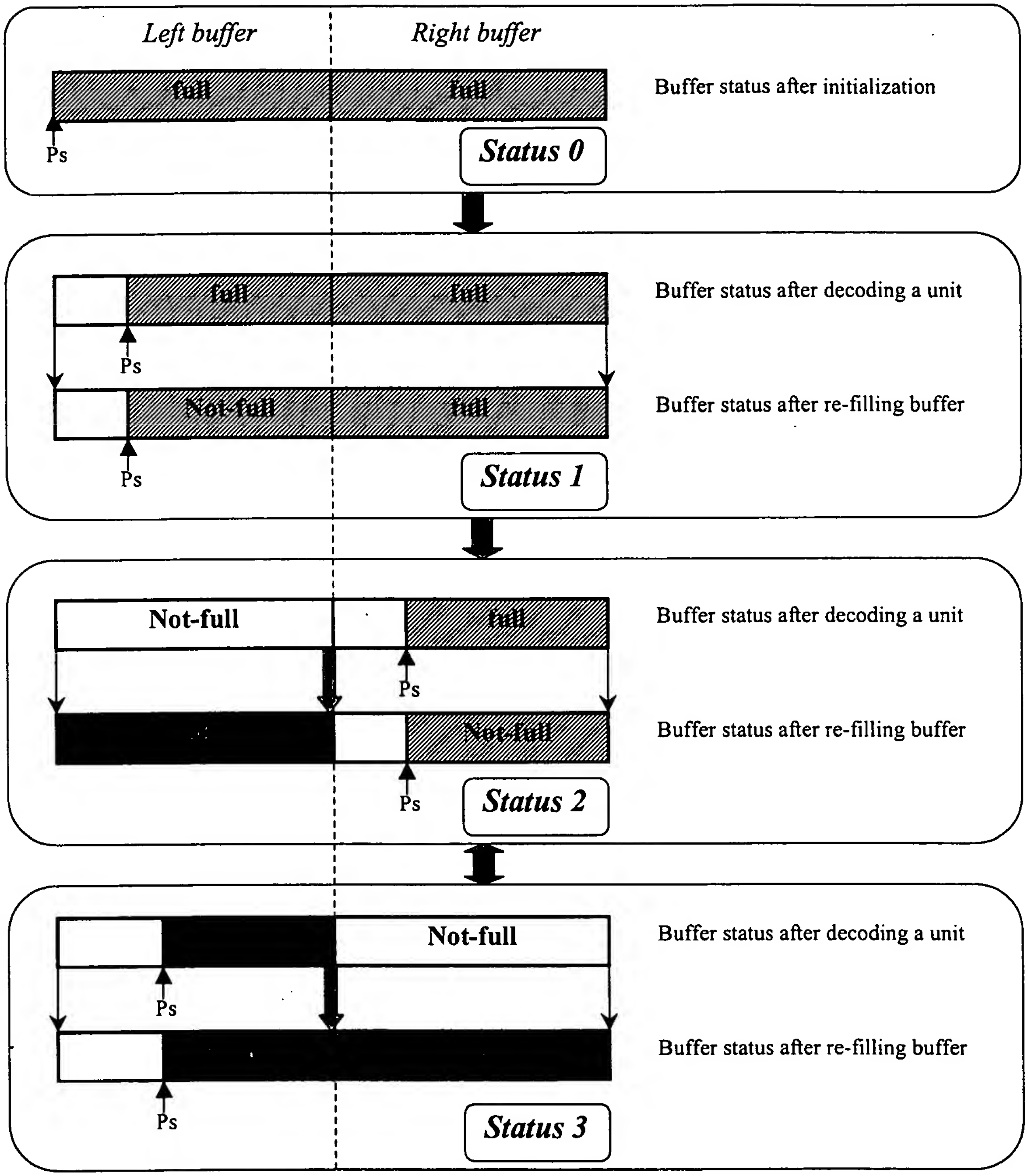

Left buffer
Right buffer
full
full
Ps
Buffer status after initialization
Status 0
full
full
Ps
Buffer status after decoding a unit
Not-full
full
Ps
Buffer status after re-filling buffer
Status 1
Not-full
full
Ps
Buffer status after decoding a unit
Not-full
Ps
Buffer status after re-filling buffer
Status 2
Not-full
Ps
Buffer status after decoding a unit
Ps
Buffer status after re-filling buffer
Status 3

Fig 52